電腦時代

COMPUTING AGE 14

微型電腦用家雜誌

HK$8.00

Dysan CORPORATION



HAVE YOU TRIED THE DYSAN DIFFERENCE?

Authorized Distributor:

Supply Consultants Ltd.
Room 1802, Valley Centre 18/F,
80—82 Morrison Hill Road, Hong Kong.
Tel: 5-8917697 Telex: 61627 SUPAV HX

DEALERS:
EXCELLENT ELECTRONIC CO., LTD. TEL: 3-313878
MICROGRAM COMPUTER COMPANY TEL: 3-728211[illegible]



MBE-420

- 15'' Impact Dot Matrix [illegible]rinter
- High Speed printing at [illegible]80 CPS
- Head size-9 pin



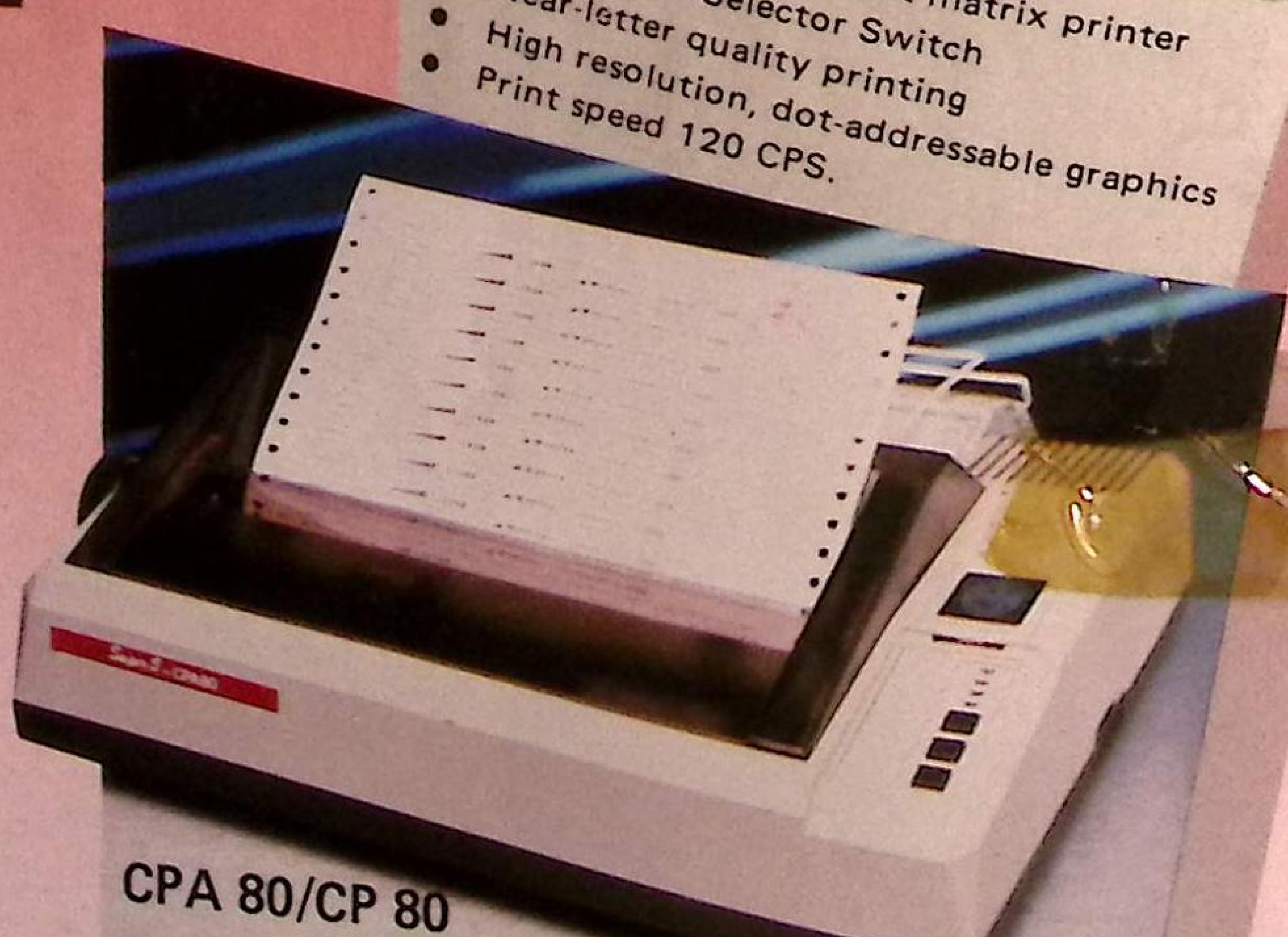
EP-1807
HIGH SPEED MULTI COLOR PRINTER IN NEW INK PLATEN SYSTEM

- New Ink Platen System in 7 colors (black, violet, blue, green, red, aqua and yellow)
- Hard copy with optional RGB video interface
- Dual mode printing at 180 CPS and near letter quality at 90 CPS
- Three-way paper handing
- 18-WIRE Print head



EN-P1091/EN-P1090
A TWO-IN-ONE PRINTER

- 9 pin Serial Impact dot matrix printer
- Print Mode Selector Switch
- Near-letter quality printing
- High resolution, dot-addressable graphics
- Print speed 120 CPS.



CPA 80/CP 80

SERIAL IMPACT DOT MATRIX GPAPHIC PRINTER

- Printing Speed 100 CPS
- Easily accessable dip [illegible]

Super 5

A & N ENTERPRISE CO., LTD.

20/F WESTERN CENTRE., 40-50 DES VOEUX RD, WEST, H.K.
TEL: 5-461179 (3 LINES) 5-488463 TELEX: 65218 MCLWW HX

IBM [illegible]

Disky®

西德專業磁碟

正符合閣下高水準要求

IBM® 私人電腦

贈



徇衆要求，將介紹期延長至一九八五年二月廿八日，凡於介紹期內購買西德 DISKY® 磁碟均可向零售商索取抽獎券，贏取巨獎。

頭獎一名：IBM® 私人電腦一部

二獎一名：Commodore® 64 私人電腦一部

三獎二十名：各得西德Disky® 5 ¼" 十片裝磁碟一盒。

Disky® 磁碟

以西德一貫嚴格工藝生產和測試

實而不華　質量可靠

歡迎到各大電腦店查詢比較

Disky is a registered trademark of Döbbelin & Boeder

Commodore is a registered trademark of Commodore Business Machine.

IBM is a registered trademark of International Business Machine.

總代理：

鷹達企業有限公司

ROCTEC ENTERPRISES LTD.

香港灣仔洛克道194-200號東新商業中心17樓　電話：5-893447(六

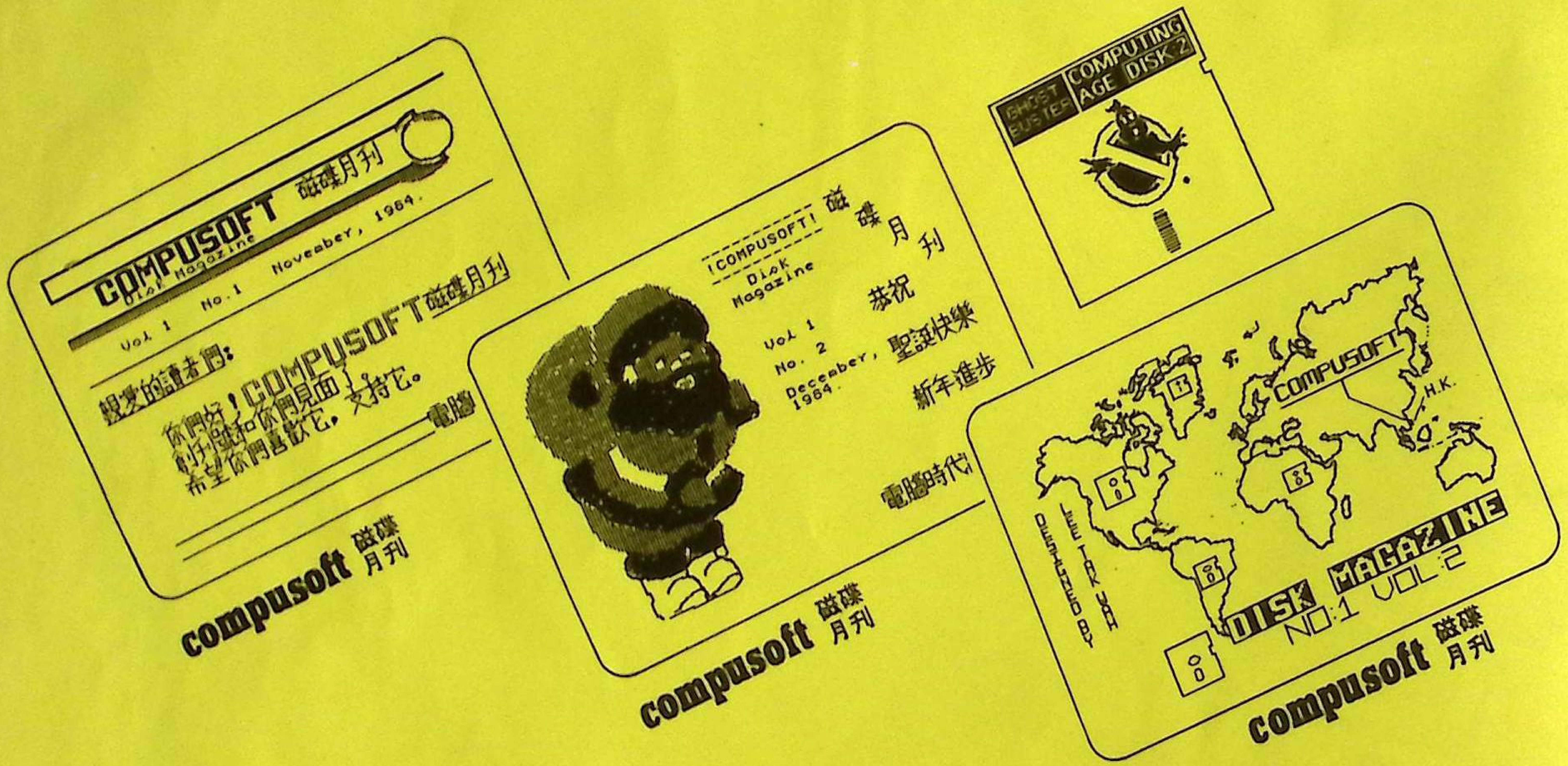
Disky® 磁碟

級電腦

請即訂閱

一本眞眞正正的電腦雜誌
蘋果電腦用家絕不容錯過的雜誌

COMPUSOFT 磁碟月刊經已出版



訂閱磁碟月刊　小小的代價！　多多的得益！

電腦時代出版社爲蘋果電腦用家的劃時代貢獻，只要你是蘋果電腦用家，就必定要擁有COMPUSOFT 磁碟月刊。你可從磁碟月刊裡找到其他電腦雜誌絕不能做到的內容。COMPUSOFT 磁碟月刊內容取材廣泛，包括教育軟件，實用工具程式，電腦自學文章，最新軟件出版消息，螢幕顯示表達技巧，精彩娛樂軟件 ，ADVENTURE 破解方法等等，適合電腦發燒友，未來電腦從業員，學校教職員，學生，關心子女學業的家長及有志投身電腦軟件工業的人士。

COMPUSOFT 磁碟月刊是一本眞眞正正的電腦雜誌，每期內內容必定的能滿足你的需求，而磁碟月刊本身用來表達內容的技巧，更是我們一羣專業程序設計師精心泡製的成果，有極大的研究價值。今日就請你立即訂閱。訂閱COMPUSOFT 磁碟月刊，只要花小小的代價，就獲得多多的好處。

COMPUSOFT磁碟月刊每月出版一次，每次出版兩張雙面灌錄的磁碟，以DOS 3.3 FORMAT，適用於任何蘋果電腦或相容的機種。

COMPUSOFT磁碟月刊只設長期訂閱，不公開發售，請立即塡妥訂閱表格寄來。

訂閱辦法

訂閱COMPUSOFT 磁碟月刊辦法簡單，訂戶每期收到磁碟後，自行COPY一份，然後寄回磁碟給我們，我們便會在下期出版之時繼續寄出。當然，訂戶亦可於每月出版之時前來電腦時代出版社讀者服務部攞取每期月刊。

讀者如對訂閱辦法有任何疑問，可致電3-312007查詢。

海外讀者只接受連同磁碟的訂閱，請附加空郵費用20%。澳門讀者亦只接受連同磁碟的訂閱，請附加多10%郵費。

只適用於本港訂戶

本人欲訂閱 COMPUSOFT 磁碟月刊

☐ 12期　HK $180(退回磁碟)
☐ 12期　HK $540(連同磁碟)*
☐ 6期　HK $100(退回磁碟)
☐ 6期　HK $280(連同磁碟)*
☐ 1期試閱　HK $ 50(連同磁碟)*

由第____期開始

支票收款人請填寫 "Computing Age Publisher"

請寄九龍中央郵箱71193號

信封面請註明「訂閱」磁碟月刊

＊磁碟歸訂戶擁有

姓名：____________________

____

地址：____________________

____

____

電話：____________________

附上劃線支票 No. ____________

（請用英文填寫，方便電腦處理）

NO.____________
Official Use

第一期內容

陽曆陰曆日子對換表

一個輸入西曆日子便可查到相對的農曆日子工具。農曆日子以天干地支來表示。

音樂火花—龍的傳人

作者利用豐富的高解像畫面襯托出一首美妙動人的樂章

五骰遊戲

可以兩人玩或和電腦做對手的擲骰遊戲，除了要靠彩數外，還要憑機智果敢的抉擇。

記憶能力的考驗

需要高度集中精神來玩的一個遊戲，玩者要憑記憶力來找到一對對的搭子，有多種難度供升級之用。

物理學的氣壓定律

特別爲中學四、五年級的同學而編寫的一個實用程式，讓同學們能溫習所學習的氣壓定律。

電子電話部

這個電話部的特點是在檢索人名地址時，非常之快，而且使用十分容易，一用即會。

波子機彈子原理模擬

讀者來透過這個程式從中獲得設計彈子反彈反應的原理，在編寫有關的程式時，可收事半功倍之效。

快速抄碟程式 COPY ＋

使用這個拷貝工具，單用戶磁碟機的讀者只要換三次碟即可將一隻沒有保護的程式拷貝來一份。

將 COPY A 及 DISK ZAP 改抄 40 軌磁碟

四十軌磁碟將會大行其道，一張四十軌的磁碟較３５軌的多出１５％的記憶空間可用。但COPY A 或DISK ZAP 本身是不能進行４０軌的拷貝，本文教你如何改動程式的來適應需求。

CONTROL RESET 使用方法

RESET 鍵是蘋果鍵盤中最特別的一個鍵，在使用上，大家或許未能充份掌握，本文作者利用多個實例來闡釋這個鍵的用途。

魔水晶二號破解方法

魔水晶二號的原作者共花了７２步來尋求魔水晶，但本港一個電腦迷，比作者行少１步便找到魔水晶，証明本港實是有一流高手的存在。

美國最新出版軟件榜

大家必然對蘋果軟件的發展很關心。這個資料榜將近五個月的蘋果軟件名單滙集一起，讓大家了解及認知。

電子郵箱

爲讀者而設的一個詢問信箱，讀者可利用此信箱來信詢問各類有關的問題。我們定當盡力以最快的速度來處理詢問

數碼式電子鐘

無名的遊戲

一行程式集

商品公司廣告集

封面高解像版頭設計

尙有其他……

第二期內容

陽曆陰曆日子對算表資料輸入法

前刊號中登載的〔西曆查農曆日子〕只能查閱１９２４年至１９８６年之間的日子，以後那些年份便沒有效。事實本程式並沒有年份限制，今期我們教你如何輸入１９８７年以後年份的資料來查閱。

磁碟雜誌中文字顯示原理剖析（一）

磁碟雜誌的內容說明主要是用中文，但又不用中文咭來輔助，究竟爲何能做到這種效果，本文將會由淺入深的爲你剖析。

化學科模擬選擇題練習

爲了讓中五的同學溫習過去所學的，以準備會考，我們由今期開始爲大家提供各科的會考模擬試題練習，首先是化學科。

接連火車卡模擬

這是一個有趣的併字遊戲模擬，利用一輛火車頭在迷宮中穿梭將亂放的英文字拉併成完整的一個英文字，過程非常有趣。

高登最受歡迎遊戲精選

十二月份高登中最受歡迎的一個遊戲軟件，我們選出來供大家娛樂。

撲克遊戲之一排七

排七是紙牌遊戲中最受歡迎的一種。我們將它寫成電腦程序來玩，遊戲的難度及趣味性大大的提高。

支持老鼠仔的軟件介紹

坊間經已有老鼠出售，價錢不算貴，但目前本港有關支持老鼠仔的軟件很少，我們集了現時美國所有支持老鼠仔的軟件供各位參考。

SUPPER HELLO

這是一個有趣的HELLO程式，在CATALOG 時，它自動將各類型的檔案分類，使一目了然。

俄羅斯輪盤

大家不用去葡京，反之安坐家中便可玩輪盤遊戲。這裡最公平不過，你可試試自己的運氣。

35及40軌磁碟檔案地圖顯示器

一般的SECTOR MAP 工具只能用於35軌的磁碟，本工具則能顯示40軌的磁碟檔案地圖。

最新版本蛇之噩夢

電腦時代曾登載過一個遊戲蛇之噩夢，本刊的作者將它再改良，玩時更加精彩緊張。

用來快速COPY 40軌磁碟的COPY ＋40

我們將COPY＋再修改使之能適用於40軌磁碟COPY。40軌磁碟將成爲一種新趨勢，本工具程式值得收藏。

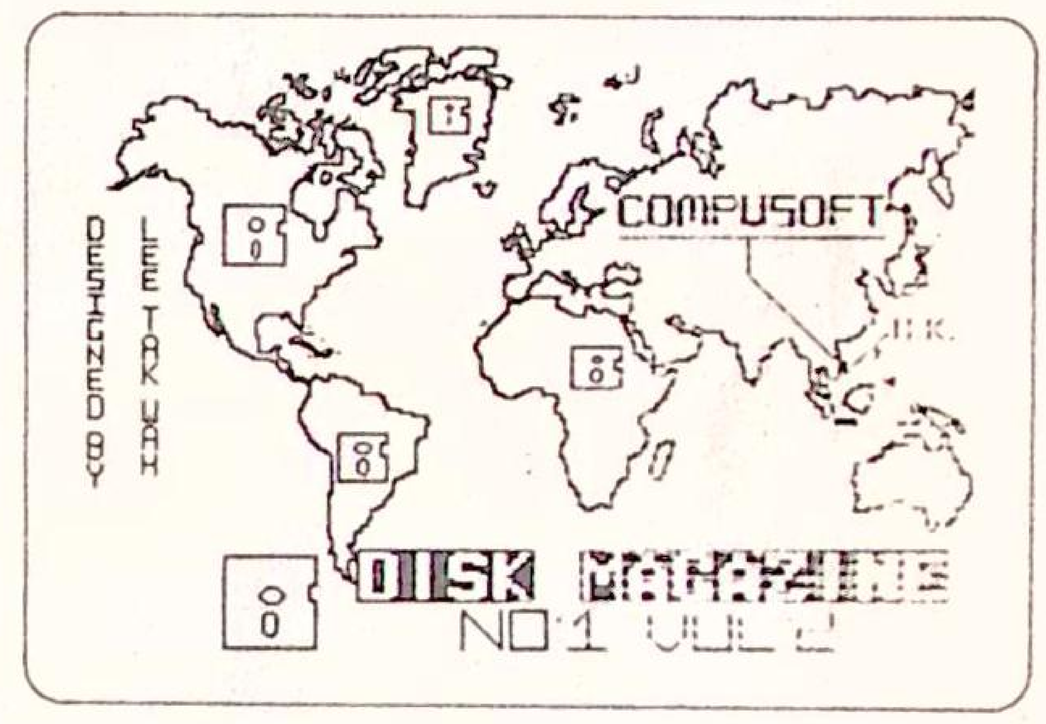
萬用數據處理庫

本數據處理庫能適用於任何情況下，絕對沒有限制，不論應用在那方面，你都會發覺它容易使用。

一九八四年全美四十大最受歡迎遊戲軟件

大家都希望知道有什麼新蘋果軟件出版，這個軟件榜將最新一個月內出版的軟件刊登，大家可清楚了解美國蘋果軟件的發展。

第三期一月底出版



大逃亡（動畫卡通大賞）

看過大逃亡這個卡通動畫，你會覺覺蘋果電腦在模擬動畫方面的能力，大家可以細心研究這個程式的編寫技巧。

陽曆查陰曆日子修訂版

這個修訂版本，加添了生肖及時辰的顯示，進一步加強本程式的能力。日後將會有一系列的程式應用這個程式，各位萬勿錯過。

你是否樂觀的人——心理測試

這是一個心理測試項目，看看你是否一個樂觀的人。

磁碟雜誌中文字顯示原理剖析(二)

今期繼續講解螢幕顯示中文的原理。希望各位能領悟其中原理，它日能發揮應用。

視窗 HELLO

第二期介紹了一個SUPER HELLO ，用法是將各類型的FILE 分類顯示，但今期這個視窗HELLO 就極之強效，能將上捲的FILENAME 不斷的再顯示，而各項執行方式（RUN、LOAD等）只要按單字母即可。

BUBBLE SORTING 原理剖析及示範

BUBBLE SORTING ：排序在程式編寫中經常會用到，但大家是否清楚了解排序的工作原理？本文提供一個非常有條理的DEMO來詳細解釋BUBBLE 排序的工作原理。導師們最宜用來講解分析給學生聽。

圍棋遊戲

圍棋遊戲是供大家訓練思考力及精神定力的好媒介。各位可與電腦來比試。

尙有其他……

體積超小

RITEMAN「威猛牌」印字機系列

適合閣下不同需求！

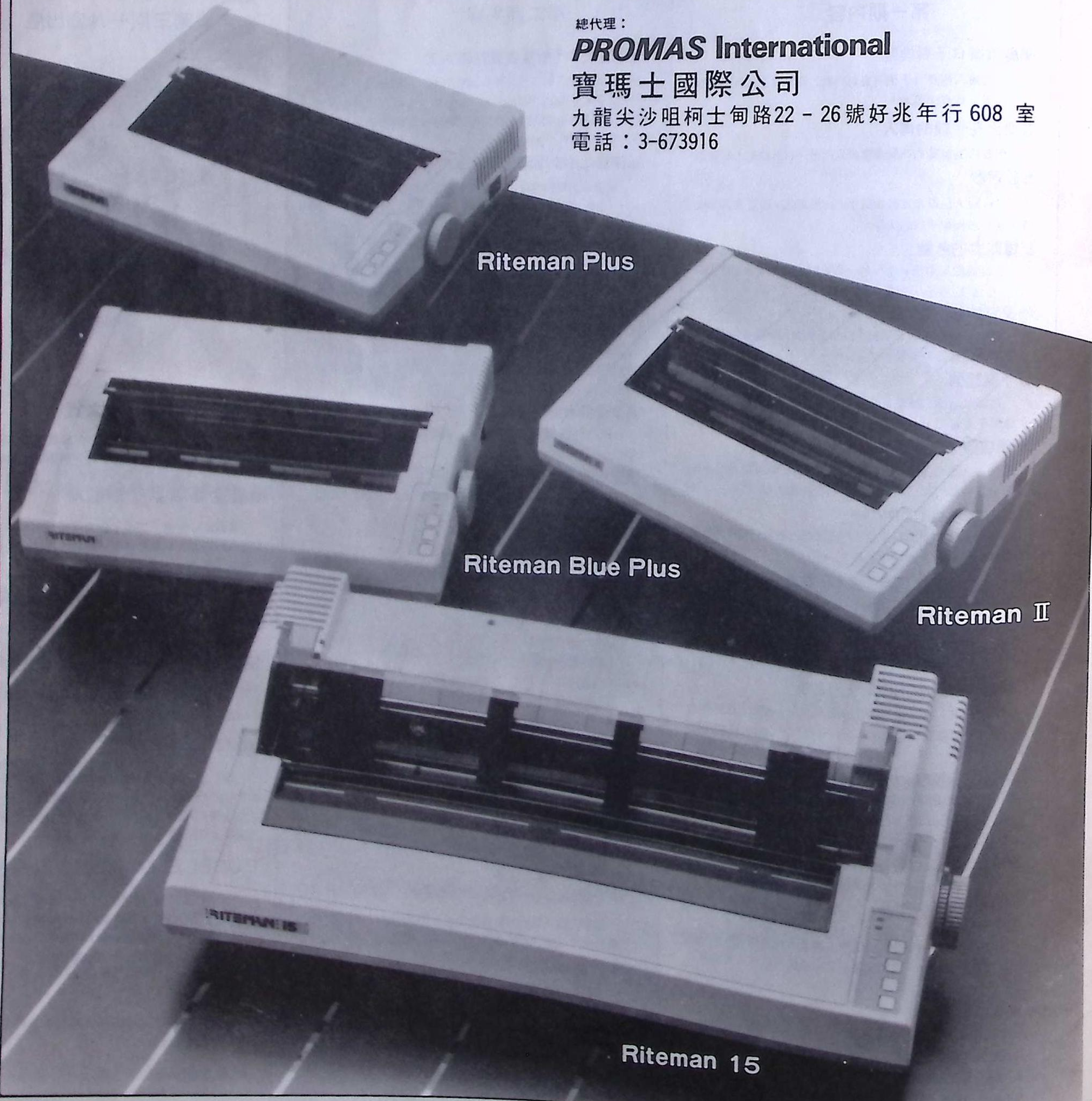
總代理：

PROMAS International

寶瑪士國際公司

九龍尖沙咀柯士甸路22－26號好兆年行608室

電話：3-673916



印字特快

- 超薄機身，體積爲普通印字機一半。
- 印字速度特快，每秒達160字。
- 多種印字功能，可印多國文字，更可作BIT IMAGE GRAPHIC DUMP(中國文字及圖案)。
- 可用軟件及硬件選擇「高質素」(LETTER QUALITY)字體，適用於文字處理。
- 採用耐用之盒式絲質墨帶，用戶可補充油墨重覆使用，操作簡易，墨帶壽命高達一百萬字，減少更換墨帶次數，不斷爲你節省金錢。

	RITEMAN PLUS	RITEMAN BLUE PLUS	RITEMAN II	RITEMAN 15
Speed(Character Per Second)	120	140	160	160
Paper Feed				
Friction Feed	●	●	●	●
Pin Feed	●	●	●	●
Tractor Feed	OPTIONAL	OPTIONAL	OPTIONAL	●
Programmable Page Length	●	●	●	●
Programmable Page Brake	●	●	●	●
Programmable Line Spacing	●	●	●	●
1/6"	●	●	●	●
1/8"	●	●	●	●
7/72"	●	●	●	●
n/72"	●	●	●	●
n/216"	●	●	●	●
Programmable Character Styles				
Combinations	66	128	128	128
Pica	●	●	●	●
Elite		●	●	●
Condensed	●	●	●	●
Expanded	●	●	●	●
Emphasized	●	●	●	●
Double Strike	●	●	●	●
Underline	●	●	●	●
Super/Subscripts	●	●	●	●
Italics	●	●	●	●
Proportional Spacing			●	●
Download Characters			●	●
Character Sets				
IBM Graphic Characters		●	●	●
American	●	●	●	●
French	●	●	●	●
German	●	●	●	●
British	●	●	●	●
Danish		●	●	●
Swedish		●	●	●
Italian		●	●	●
Spanish		●	●	●
Graphic Modes				
60 Dots/Inch	●	●	●	●
72 Dots/Inch	●		●	●
80 Dots/Inch		●	●	●
90 Dots/Inch		●	●	●
120 Dots/Inch	●	●	●	●
240 Dots/Inch		●	●	●
Quiet Mode		●	●	●
Interfaces				
Centronics Parallel	●	●	●	●
RS-232C Serial	OPTIONAL	OPTIONAL	OPTIONAL	OPTIONAL
Warranty	1 YEAR	1 YEAR	1 YEAR	1 YEAR

重要啓事

由於近日市面上出現大量水貨充斥市場，爲了避免消費者購入這些「水貨」，而失去應有之保養服務，所以本公司已在本年十一月一日起實行以下措施。

凡經本公司或各大零售商發售之貨品，箱面上均貼有「雲生保養」紅色標誌（如圖示），藉以分別「行貨」及「水貨」。顧客購買本公司貨品時應認明該標誌及向零售商索取由本公司發出之有效保養證。但在這段期間，零售商可能存有無此標誌之貨品，顧客仍可向零售商索取本公司有效之保養證，如有疑問，請即致電本公司查詢。

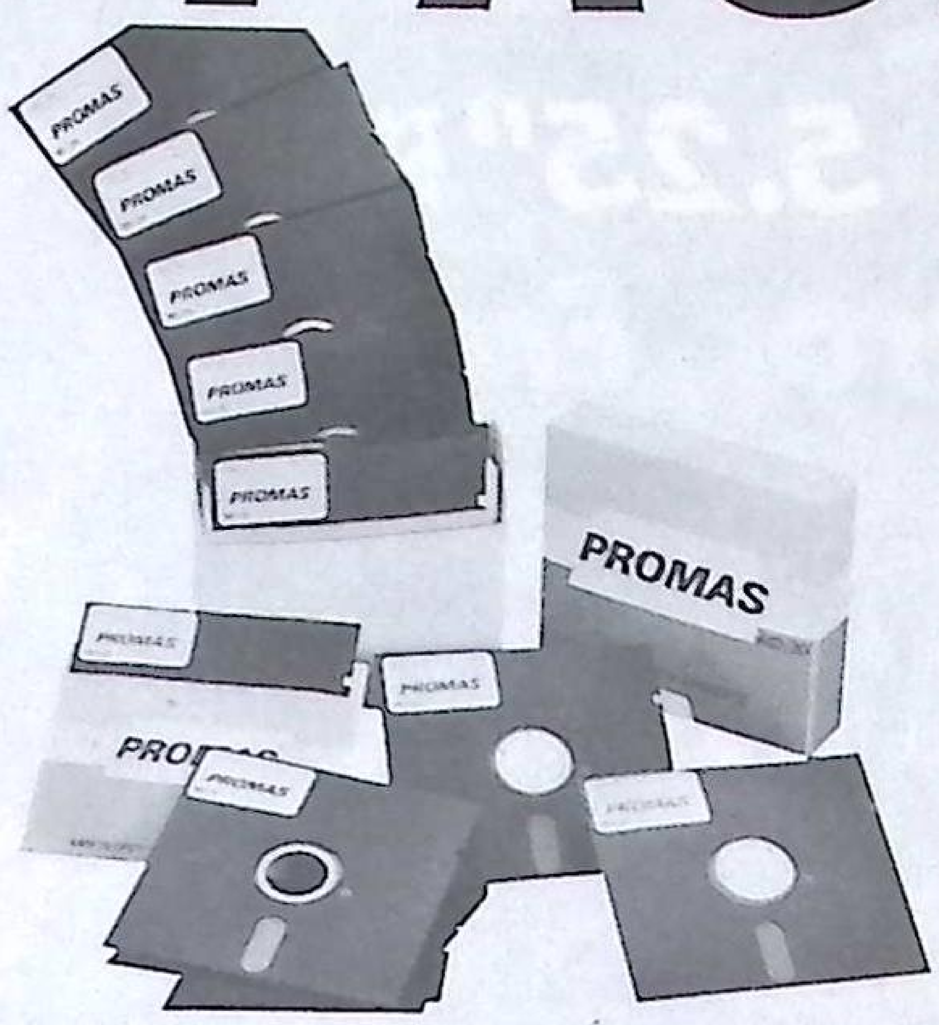
此舉乃保障消費者之權益，本公司重申對無保養証之貨品(水貨)，絕不提供任何保養。請各顧客購買時特別留意，免招至損失。

特此敬告各顧客
雲生電腦有限公司謹啟



總代理：
VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司
九龍尖沙咀柯士甸路22-26號好兆年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

PROMAS 電腦產品



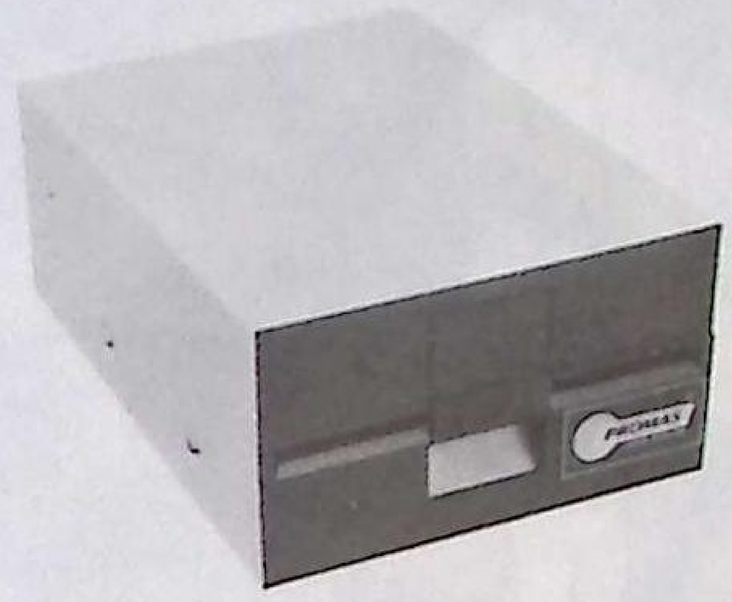
5.25″磁碟

雙面雙倍密度，儲存量達500 K，適合一般電腦使用。全日本製造。



MODEL JT 5.25″薄身磁碟機

多種「WRITE PROTECT」及「HEAD SEEK SPEED」選擇，適合不同應用。APPLE II*及相容電腦專用。全日本製造



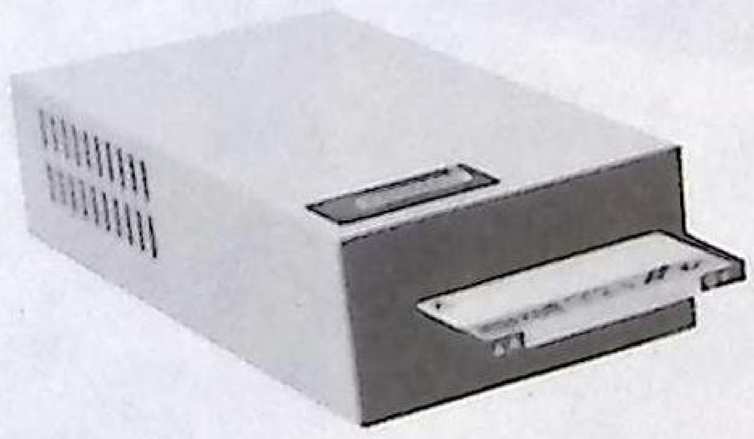
AS-100 5.25″磁碟機

傳統設計，品質檢定合格，APPLE II*及相容電腦專用。全英國製造。



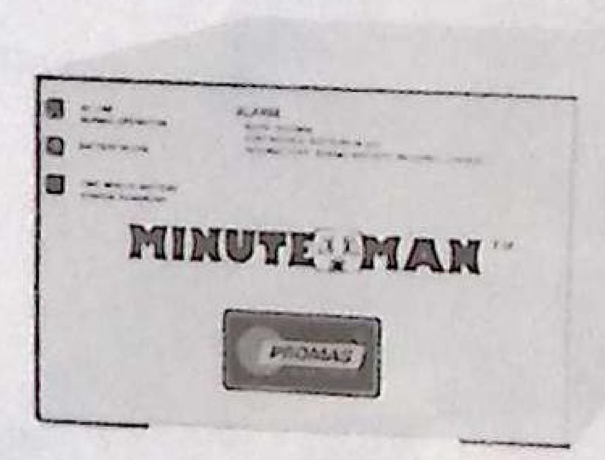
MODEL JA 5.25″磁碟機

全新設計機械部份，特別柔順。APPLE II*及相容電腦專用。全日本製造。



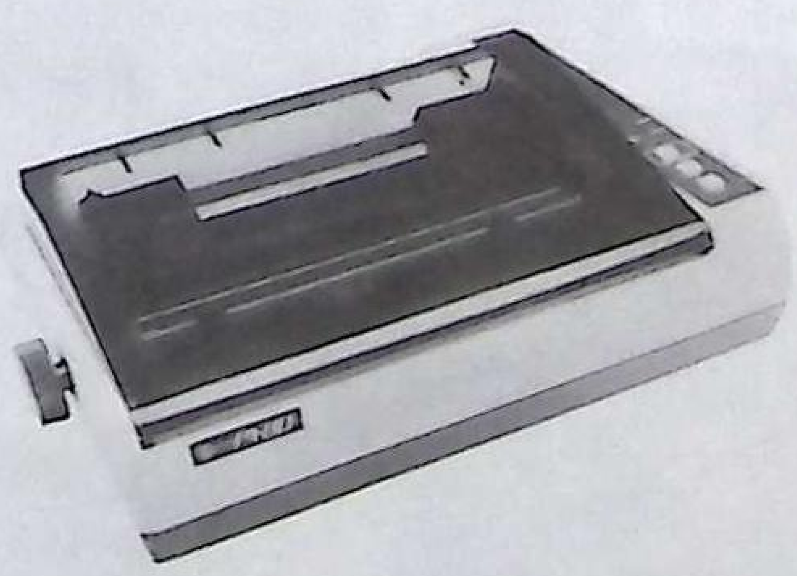
305 3″磁碟機

多重保護資料，與一般APPLE II* 磁碟機共用或可單獨使用。全日本製造。



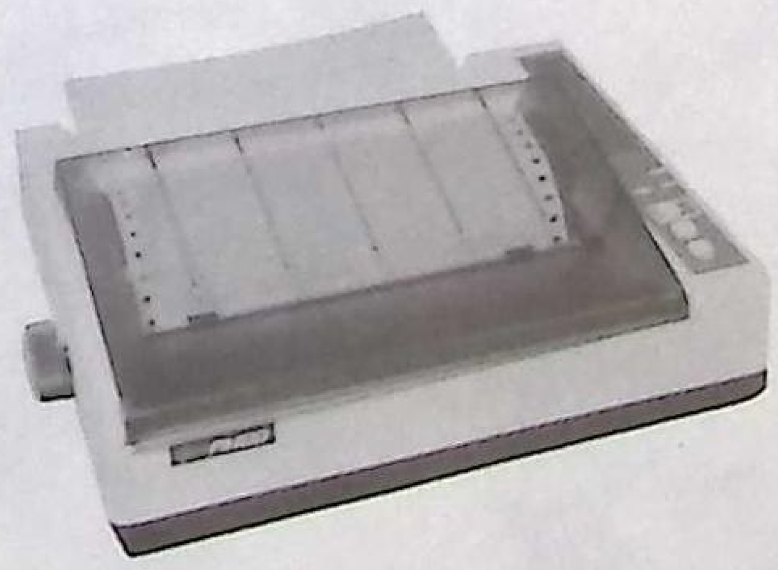
MINUTEMAN後備電源供應器

防止一般電源問題對電腦及精密儀器造成之破壞。全日本製造。



PX-80印字機

印字速度 每秒80字。CENTRONICS介面接口，適合一般電腦使用。全日本製造。



PX-100印字機

印字速度 每秒 100 字，功能與P X －80相同。全日本製造。



PF-150 手提圖文傳眞機

攜帶方便，可與世界各地傳眞機互傳訊息。全日本製造。

總代理：

PROMAS International

寶瑪士國際公司

九龍尖沙咀柯士甸路22－26號好兆年行608 室　電話：3-673916

* Apple II is a registered trademark of Apple Computer Inc.



OPERATIONAL PERFORMANCE

MODEL	Sides	Density	Sector	Track	TPI	Format	Capacity
MD/2D	2s	Double	Soft	35 or 40	48	None	500KBytes

PROMAS International
寶瑪士國際公司
九龍尖沙咀柯士甸路22－26號好兆年行608室　電話：3-673916


新設計磁碟機
PROMAS Model JT
PROMAS
MARK II
WRITE PROTECTION
Write thru write-protect seal
Normal write-protect mode
All disks write-protected
HEAD SEEK
Default seek
Slow seek
DEFAULTS:
● 獨特 "SLOW SEEK" 功能，存取資料更準確。
● 內置WRITE PROTECT及HEAD-SEEK速度選擇掣。
● 採用新設計高質素機械部份，可靠程度更高。
● 支架用純不銹鋼製造。
● 完全適用於APPLE II* 系列電腦。
● 全日本製造及質素檢定合格。
總代理：
PROMAS International
寶瑪士國際公司
九龍尖沙咀柯士甸路22－26號好兆年行608室　電話：3-673916
*Trademark of Apple Computer Inc.



停電 都冇損失！

全新設計線路
效果更佳更理想

Minuteman UPS「自動後備供電器」

保障電腦及電子器材運行正常，不受電源中斷影響而失去寶貴資料及損壞昂貴器材。

- 可用作防止電壓下降、電壓冲擊及電源中斷等，確保電源供應正常。
- 當市電停止供應時，能立即自動提供後備電源，並以蜂鳴器警報及指示燈顯示。
- 供電時間視乎配用器材之耗電量而定。
- 在市電供應恢復正常後，本機會隨即自動充電。
- 適用於任何電腦，硬碟儲存系統及精密電子儀器。
- 全日本製造，價錢比同類型產品便宜一半。

總代理：

PROMAS International

寶瑪士國際公司

九龍尖沙咀柯士甸路22－26號好兆年行608室　電話：3-673916

重要啟事

本公司自1979年成立至今，並未有設立任何分公司。近日市面上，有一電腦公司因慕名而冒名使用與本公司相似之名稱，售買「假貨」及訛稱由本公司負責維修保養。

本公司一向不售賣任何「假貨」及無其它分號，故此請各消費者加倍小心，免被漁目混珠，招至損失。

本公司並不排除對該等不法商人採取一切法律行動。敬請認明本公司之英文全名串法及爲有限公司。

VINCENT COMPUTER CENTRE LTD.
雲生電腦有限公司
九龍尖沙咀柯士甸路22-26號好兆年行608室　電話：3-7211288-0　電訊：HX 38316 VCC HK

各位業餘程序編寫員萬勿錯過本文

電腦時代爲你開闢一條新途徑

各位業餘程序編寫員，你們的軟件創作才華有沒有被人忽視？你們的心血作品有沒有人賞識？

我們知道，你們這群業餘程序編寫員絕大部份都是天才橫溢，所創作編寫的軟件並不遜色於任何的職業程序設計師，而且，有許多更是超乎市面要求的商業水準。你們的作品之中，有許多是經年累月的經驗。我們懇請你們不要埋沒自己的天才，將你的經驗創作成果與每一個電腦迷分享。

電腦時代出版社與 COMPUHOUSE 軟件製作所願意爲你全盤策劃出版你的軟件創作成果，讓它流傳到每個電腦用家手中。你絕對不用担心任何的問題，我們會代你効勞一切有關版權、出版、發行製作等問題，你只要安心在家創作便可。

我們擁有豐富經驗的行政人員、軟件工程師、編採出版人材。而且更具有一個廣大的市場發行網，我們的發行網除遍達香港外，還推廣到台灣、星馬泰等東南亞國家，而且更延伸至美國及加拿大。中國大陸這個龐大的市場當然也在我們足跡之下。

透過我們出版軟件，你將會得到最大的收穫。

我們龐大隊伍的生產綫，不單設備完善，且有足夠能力應付任何各方面的需求。

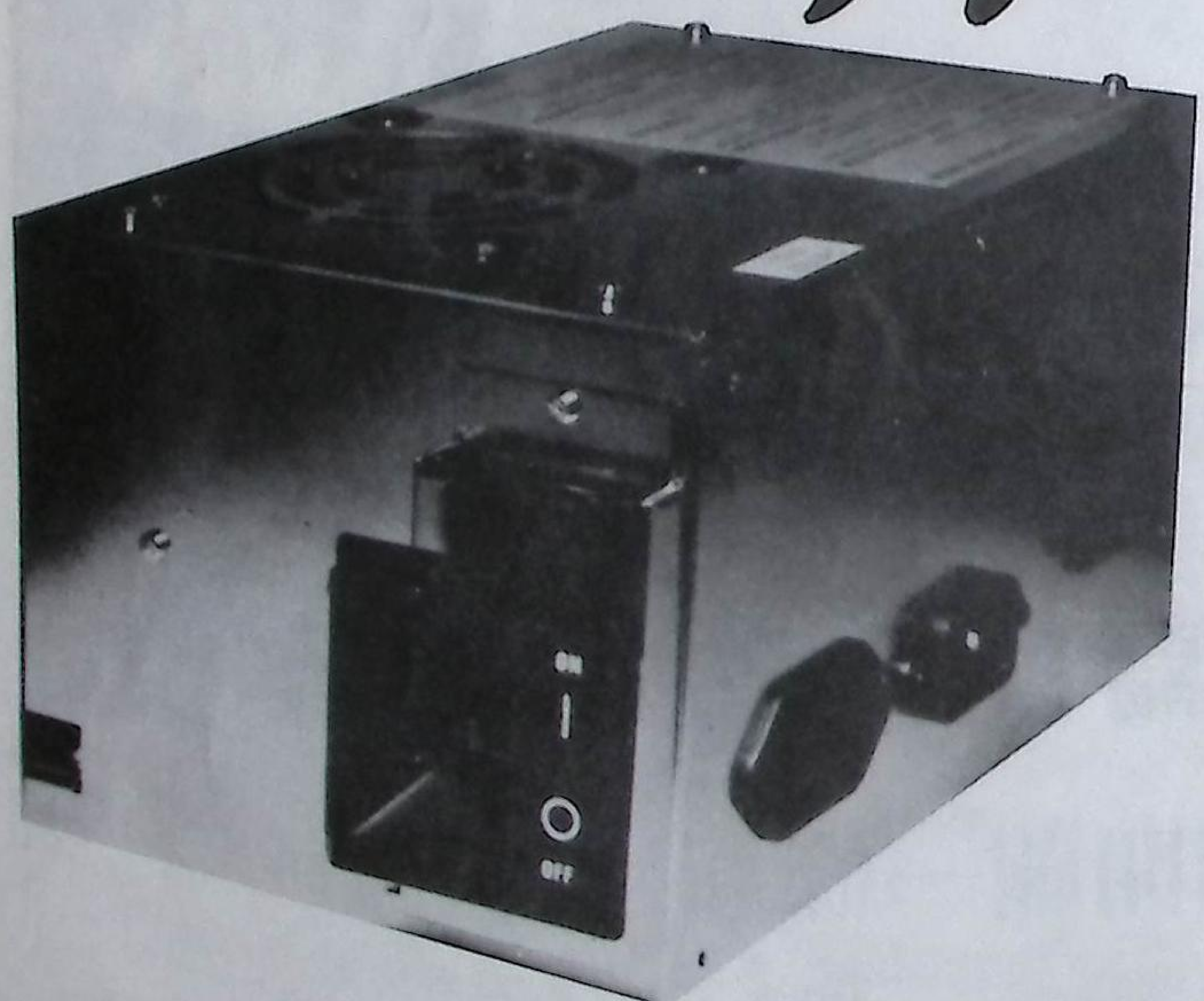
我們歡迎任何機種及類型的軟件創作，無論是ＩＢＭ、ＡＰＰＬＥ電腦、由娛樂軟件、教育應用至商業軟件，只要你們有信心認爲自己的作品有一定的商業水準，我們都樂意爲你出版。

歡迎你們携同作品前來本出版社商談，或電3—312007 與譚先生聯絡。

LEE'S ELECTRONIC CO.

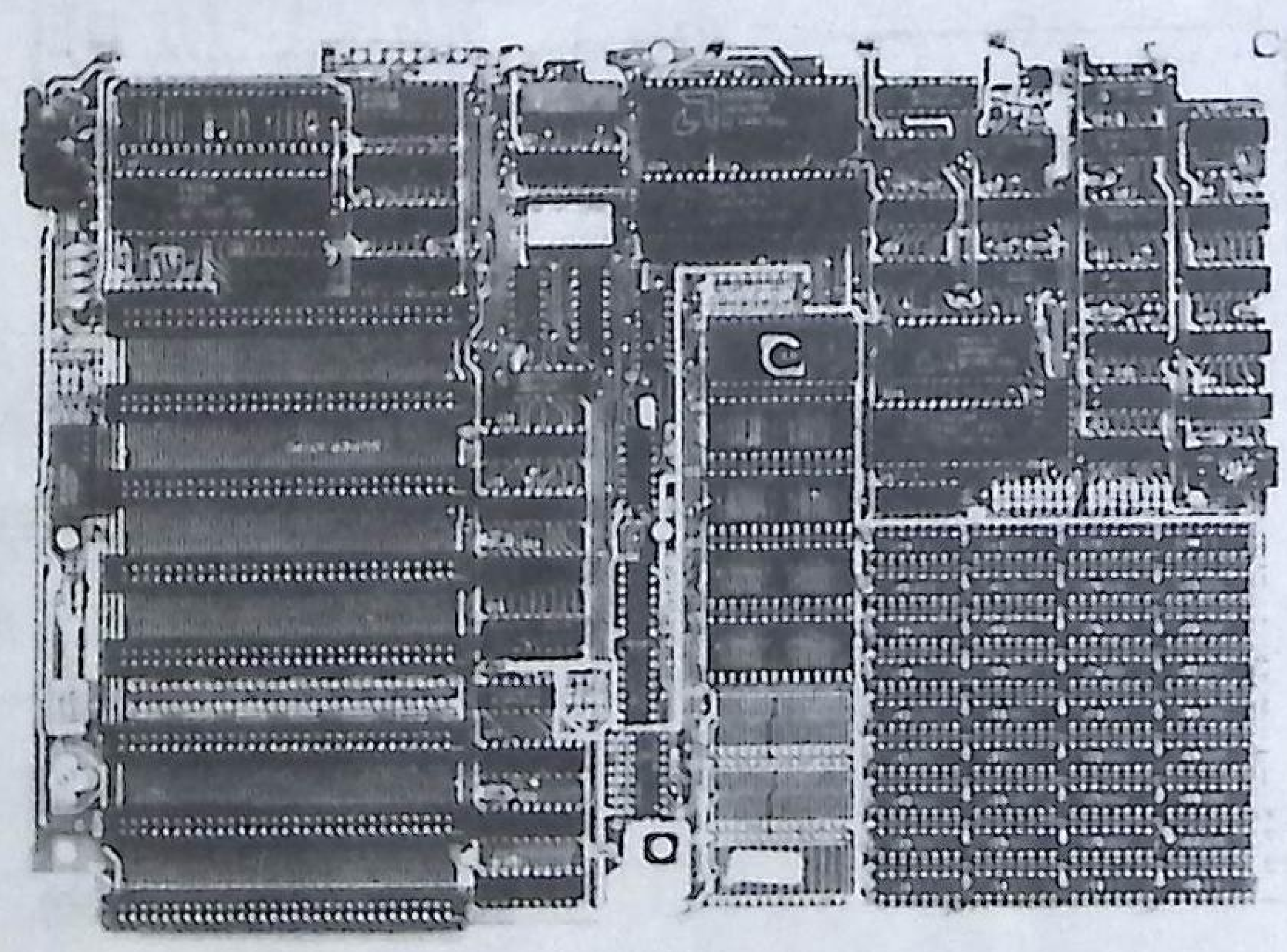
李氏電子公司

130 W



Made in Japan. PC Switching Power Supply

+ 5V, 12A	+ 12V, 5A
– 5V, 0.5A	– 12V, 0.5A



Mother board

* 4.77 MHz 8088 16-bit microprocessor with 8087 co-processor (optional)
* 64K RAM on board with partity check and expandable to 256K
* ROM 8K by te Super XT BIOS expardable to 40K by te
* 4 channel DMA 8237
* 8 channel interrupt 8259
* 8 I/O slot

九龍深水埗黃金商場地庫66A室
電話：3-7291022

Basement Shop No. 66A, Golden Building Shopping Centre, 146-152, Fuk Wah St., Sham Shiu Po, Kowloon.
TEL: 3-7291022

香港北角英皇道255號國都商場3樓
213室　　電話：5-718392

Shop No. 213 Olympia Shopping Centre, 2/F., 255 King's Rd., North Point, H.K.
TEL: 5-718392

ACCUTRACK

滿100分之磁碟
享譽世界，品質取勝

100%無差錯。
100%確保電腦資料完整，萬無一失。
100%耐溫與耐濕。
100%壽命特長，終生保用。
100%不損傷磁頭。
100%靈活運轉。
100%適合各類型電腦。
100%美國原廠製造。
100%安全使用。
100%值得你信賴的高質磁碟。

現擧行特價優待大贈送，凡剪下印花到任何電腦公司購買ACCUTRACK磁碟壹盒，可節省HK$5.00
（每盒磁碟祗能使用一枚印花，有效期至1985年2月底止）。

HK$5.00

C

ACCUTRACK
accurate tracking



Manufactured in U.S.A. by Dennison Computer Supplies Inc.

特約經銷商：

資庫電腦系統有限公司
香港中環德輔道中21號
歐陸貿易中心601室
5-230645

高寶電腦系統有限公司
九龍尖沙咀
柯士甸路20號703室
3-7238730

恆溢電腦公司
九龍油麻地吳松街150-164號
寶寧商業中心41-42號地下
3-7243778

Exclusive Distributor:

PEC

PASONIC ELECTRONICS LTD.
1601 Chung Kiu Commercial Building, 47-51 Shang Tung Street, Kowloon, Hong Kong.
Tel: 3-306636, 3-306627 Telex: 56076 UNIMY HX

令蘋果發言的最佳工具
MOCKING BOARD專用的

SPEECH WRITER

MOCKING BOARD 博士

自蘋果電腦推出TALK CARD 一說話卡之後，曾經掀起了一股的「說話：熱潮，但可惜由於TALK CARD所產生出來的聲音是比較粗劣和失眞率高，此爲一般蘋果玩家所失望呢！熱潮亦隨之消失了。

最近，九龍深水埗的電腦商場已漸轉爲一個音响商場，這不是電腦商人已轉營音响行業，而是商場已流行一張專爲蘋果電腦而設的MOCKINGBOARD電子擴展板。

它是一張可產生高度原音的身歷聲音樂板，隨硬件附上的SOUND UTILITY 磁碟，更容許用者以非常簡單的程序去產生所需音樂資料，並且發聲副程式是很容易加進任何程式中（包括機械碼程式，方法可參閱13期的MOCKINGBO—ARD 介紹文章）。特別的電子板結構，使發聲過程與6502同步程序，增加了程序的效果，和所產生的音韻却遠比蘋果原有的SPEAKER之單音更爲完備與幻化多端。

既然MOCKINGBOARD具有高度的身歷聲效果，而人類的聲音也是由音韻所組成。所以，利用MOCKINGBOA—RD來模擬人類的聲音却是綽綽有餘呢！

可惜，MOCKINGBOARD附同的SOUND UTILITY 軟件套裝中，却缺少了很多應用常式，所以用者必須下一番心思才可完全發揮MOCKINGBOARD的高超效能。

因此，筆者特別精心編寫了幾個常用程式，推薦給讀者們，並藉此推廣「電腦說話」的時尙。

在原庄的SOUND UTILITY 的磁碟中，有足夠空間來儲存一些常用程式和程式所產生的檔案，但必須緊記先要將原庄版本的SOUND UTILITY 程式抄錄於其他後備磁碟上，以避免資料的流失。

筆者所編寫的常用程式就像一個文字處理器（姑且稱之爲SPEECH WRITER），用者可通過此程式輸入或編輯一些文字資料。其與文字處理器程式的最大分別便是將文字資料以MOCKINGBOARD 讀出，而不是印出於印字機上。

基於以上的種種原因，程式便不需着重文字的顯示，而反過來却是字眼讀音的準確性！當用者完全鍵入和執行程式1後，螢幕便會顯示出一個程序選擇表。通過適當的程序，用者是可控制MOCKINGBOARD 將部份或所有文字資料讀出，甚至修改文字內容或將一些文字檔案併合及讀出。文字編輯功能變化多端，操作亦非常簡易，實可媲美一般文字處理程式。

SPEECH WRITER 常式所用的變數已列明於表1中，而且常式內的副程式均有備註說明程序功能，我亦非常歡迎各同好能動手改良本程式的功能。

行120至150是TEXT—TO—SPEECH 副程式，也是此程式的靈魂所在，因爲它是讀入SOUND UTILITY 的部份機械碼（將文字轉爲發音參數）程式。

若程式執行至此，而電腦又送出FILE NOT FOUND 的錯誤訊息時，這可能是用者未將SOUND UTILITY 程式磁碟插入磁碟機中，又或是一些電腦商將SOUND UTILITY 內的副程式改了名，所以便導致程序錯誤，各位請加留意。

要解決以上難題，用者衹須修改行120至150 的讀入檔案名稱來符合磁碟上的便可。

輸入演詞

在程式的主選擇表上，用者衹須按下「1」字便可進入文字輸入副程式，但用者必須注意此選擇是代表用者要重新建立一個新的文字檔案，所以在處理器內的舊有文字資料，便會完全被清洗。

在文字輸入狀態下，用者也有一定的輸入格式限制，如每次輸入字串不能長過255個字，而且不可是「，」豆點和「：」冒號等。但用者不須灰心，因爲其文字資料並不是用來看，而衹是「讀」出。

每次的文字輸入，程式也會提供修改文字內容的機會，所以操作上應該沒有問題。基於英文字母的組合千變萬化，一些比較複雜的英文字句，程式中的TEXT—TO—SPEECH 副程式亦未必能正確地將其讀音解碼。同樣，即是一些裝有模擬發聲的16 BIT 電腦（如TI—PROFESSIONAL和D—EC TALK 等），也不能完全解決以上的難題。

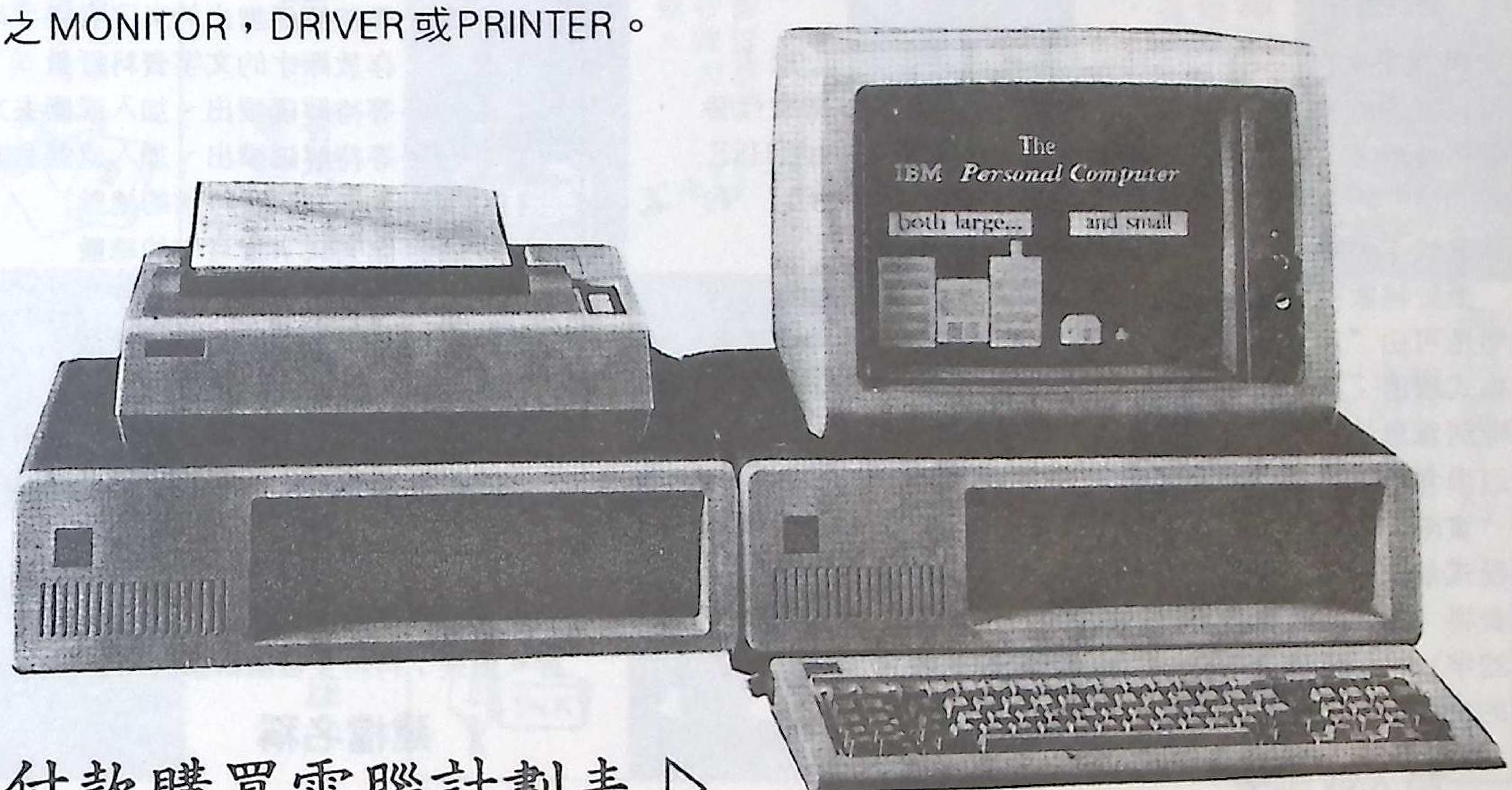
在這樣的情形，我建議各用者不妨利用拼音的方法來代替英文字句，反正這些字眼是衹由電腦讀出來。比方MOCK—INGBOARD 的英文字句，是由2個英文字組成，而且讀音也是連貫的。若以程式輸入原本字眼，而由MOCKINGB—OARD讀出，你可能發覺效果未如理想。因此，細究MO—CKINGBOARD字眼的拼音—CARIBBEAN CRUISE，它可由以下的文字資料CARE IB BUNCROOZ 來代替，那麽你會發覺電腦可讀出接近完美的發音了。當然，以上衹是一些個別的例子和挑剔的測試，而TEXT—TO—SPEECH 副程式是有足夠能力將大部份的英文字音原本地讀出，效果也比TALK CARD 還要好幾十倍呢！

還有，程式對每行文字資料解碼和讀出時均有適當的暫停時間，所以聽來也感順適調和。程式中的文字資料輸入部份是由APPLESOFT的INPUT指令達到，所以用者在輸入一些重覆的句子或字眼時，可利用正常程式編輯鍵（先按入

現代電腦訓練中心
香港電腦推廣協會
合辦

全港首創—分期付款購買電腦辦法

可購買 IBM，EAGLE，EVERGO，PC-88，APPLE-IIe，COLUMBIA及APPLE-II與IBM之相容機種。亦適合購買上述電腦之MONITOR，DRIVER或PRINTER。



◁分期付款購買電腦計劃表▷

電腦或任何配件之價錢	分期付款每月供款數（免首期） 9 mth	12 mth	18 mth
$ 2,000	$ 272	$ 209	$ 147
$ 5,000	$ 679	$ 523	$ 368
$ 7,000	$ 950	$ 732	$ 514
$10,000	$1,358	$1,046	$ 736
$15,000	$2,036	$1,568	$1,104
$20,000	$2,716	$2,092	$1,472
$30,000	$4,072	$3,136	$2,208
$50,000	$6,788	$5,228	$3,680

附帶利益：—

①免費代客選購合用電腦。

②免費安裝。

③低價代客編寫程式。

④免費贈送電腦課程。

⑤免費贈送現成電腦Program。

⑥一年免費保養。

地址：九龍深水埗福華街156號黃金商場地庫20號及54號

電話： 3-601531 3-869085

令蘋果發言的最佳工具 SPEECH WRITER

ESC，繼而I，J，K，M 和左右箭咀鍵等）來重覆，螢幕上的文字。若你是電腦初哥，那麽筆者提議你最好以常規輸入資料了。每次的文字輸入後，電腦是會將整句文字重現於螢幕，以供檢錯。

最後，若用者已輸入所需的文字資料後，打入「！」符號並按RETURN ，便可完結文字輸入程序和跳回程序選擇表上。

縮寫

TEXT—TO—SPEECH 可算是一個具智慧的副程式，因爲它容許用者輸入一些英文字的縮寫或其他單字母來代替，這可減短文字資料輸入的時間和記憶體的浪費。如THRE—ESOME 的英文字是可縮寫成3SUM或由〝N KSD〞 的拼音取代；又如〝CUE〞可寫成Q呢！

至於純數字的字串，如ONE NINETY SEVEN 的英文字句是可由〝19T7〞 取代。同樣138 的數字是可以英文文句格式讀出，其輸入格式祇須爲〝1 THIRTY 8〞便可。一些特別意思的符號，如％、＆和@等均有設定的讀音參數，所以用者不須費心其讀音，而直接輸入符號便可。

實際上，除了SPEECH WRITER 佔用了部分記憶體外，程式是預留了約13K BYTES 的記憶空間來存放輸入的文字資料（最多爲50行（每行255 字）；又或400 行（平均每行32字）），這已足夠一般用家利用電腦來讀出一篇悅耳動聽的文章呢！

加入或删去

剛才也曾說過程式祇容許50行（每行255 字）的字串輸入，所以每加入一句文字資料行後，你必須預計處理器內的資料容量。不停地加入過長或大量的文字資料，可導致程式與變數和字串列陣重叠，造成了資料流失和HANG 機的現象（正因蘋果電腦的記憶系統中，程式是由低記憶處向上鏈，而變數和字串列陣則由上而下擴展，所以過量的列陣可遮蓋了程式的最尾部份，以致系統失靈）。

唯一解决以上的辦法，便是空按 RETURN 鍵（即輸入空的字串行），直至程式告訴你已沒有空間來存入文字資料爲止，從而估計所餘的列陣數目。轉過頭來，你可删去剛才的空字串句，而加入你所需的文字資料行。

同樣，删去文字資料行時也要注意所删的行數，而每删一行的文字句，其後隨的文字資料行數便會相應減一，若用者輸入的行數是少於 1 或大於行數的總和，程式是會檢出並顯示BAD LINE NUMBER的錯誤訊息。

最佳的編輯資料方法，便是每到一定數量的資料輸入後，來作一次資料儲存於磁碟檔案上，這也是一般經驗程序操作員的良好習慣。

檔案合併

當用者建立了讀出文字資料檔案後，是可隨意修改或在檔案末端再加入新的文字資料。過程非常簡單，用者先選擇第＃ 4 項讀入資料功能，從磁碟檔案讀取資料於編輯器內，再按功能＃ 7 電腦便容許你在檔案末端再加入新的文字資料行。最後，再將整個編輯器內的資料儲存於磁碟上去。

表(1)　變數表

D＄—CHR＄（4）	是驅動磁碟的控制碼
IN＄	—文字資料輸入變數代號
IN％	—數字輸入代號
TL	—文字資料句的總數
TL＄	—文字資料字串列陣
ML	—最多的文字資料句的輸入數量
MB＄	—等待解碼讀出的文字資料字串
LN	—存放剛才的文字資料行數
L 1	—等待解碼讀出、加入或删去文字資料群的首行
L 2	—等待解碼讀出、加入或删去文字資料群的末行
IL	—加入文字資料句的總數
DL	—删去文字資料句的總數
VT	—游標定位參數
SL％	—永久字串的低位元指標
SH％	—永久字串的高位元指標
LO％	—清除字串列陣的低位元指標
HO％	—清除字串列陣的高位元指標

建檔名稱

程式是容許用者建立任何內容的文字資料，甚至任何檔案名稱（包括以數字組成及 1 至25個字串長度的檔名）。其實，程式在建檔的部份中已加上了MOCK 英文字母作爲所有文字檔案的首 4 個字，原因亦相似MAGIC WINDOW 所產生的檔案名也有MW 2 字的結尾，目的是將一些屬於MO—CKINGBOARD 的檔案分類，而用者在輸入所需讀入的檔案名稱時，請不要多加MOCK 首 4 個英文字，因爲程式內部已自動加上呢！

應用方面

剛才我已簡略說明了SPEECH WRITER 的操作過程和功能，而完成了建檔過程後。用者是可進一步利用此程式來作某大會上的發言代表，相信它清晰悅耳之發言可使一向沉悶的會議變成生氣勃勃呢！

開動蘋果電腦（當然MOCKINGBOARD 是必須的週邊），讀入SPEECH WRITER 程式和所需的文字檔案，再按功能選擇＃ 2（將編輯器內的文字解碼並讀出）。在進入功能選擇＃ 2 時，用者也可選擇讀出整篇或部分文字行。

還有，若用者能多花一些心思於本程式中，改良要求用者選擇程序的部分，亦不妨加入一些已編排好的文字資料來驅動MOCKINGBOARD 發出聲音去指導一些從未使用過電腦的初哥去執行程序。

在此，我提議你們嘗試從行4000着手改良整個程式。首先，建立一個程序指引的文字資料檔案（如名爲〝MESSA

香港教育司署立案（1957年創校）

世界函授學校

規模大　優點多　科目好

初級電腦科
電腦基本原理，電腦操作，邏輯電路，程式編寫，八個月畢業。

初級無綫電科
全部圖解教導，原理裝修並重，極易學習，十個月畢業。

半導體裝修科
配合進步的技術，具初學或職業的裝修人員必修的學科，十個月畢業。

實用數理
可以加强數理基礎，費用少，收效大，6個月畢業。

x+y=z

實用商科
著重實際應用，加强商業學識，期間短，學習易，六個月畢業。

校址：
香港九龍彌敦道五六六號9樓A座
電話：③－803532

講義
全部經教育司審定，印刷精美，閱讀容易，免費供應。

師導
國內外著名大學學士，經香港教育司考核合格，教導認眞，決不馬虎。

無綫電工程
設計，研究和深造的理想學科。一年半畢業。

汽車工程
適合職業及業餘人員加强對汽車方面之學識。

冷氣工程
冷氣，冷藏人員缺乏，專門學校極少，一年半畢業。

商業美術科
內容新，範圍廣，包括美術和最新廣告設計。一年畢業。

應用漫畫
漫畫可訓練頭腦，增强思想，取材新，學習易，六個月畢業。

INK

用費
除低廉之學費外，學生決無額外負擔。

文憑
本校頒發文憑，均經政府簽証，實是最佳學歷証件。

本港學生可利用電話詢問索取章程　電話③－803532

請依綫剪下，填好姓名，回郵地址，貼在信封上投寄，本校即免費寄上簡章

郵件請寄：P.O. BOX 70011, KOWLOON, CENTRAL POST OFFICE
九龍中央郵政局信箱 70011 號

技術就是財產

（寄自）

姓名：

地址：

（請註明要學的科目）

☐初級電腦科
☐無綫電工程科
☐初級無綫電科
☐半導體裝修科
☐實用數理科
☐汽車工程
☐冷氣工程
☐實用商科
☐商業美術
☐應用漫畫

請貼上郵票

香港教育司署立案

To：世界函授學校註册處

P. O. BOX No. 70011
KOWLOON CENTRAL POST OFFICE,
HONG KONG.

香港九龍中央郵局信箱70011號

電話：K-803532

學問就是力量

令蘋果發言的最佳工具 SPEECH WRITER

—GES”），接着刪去行4000、4010、4020和4060。再加上上行4000 N $＝“MESSAGES”和行4075 DIM TL $（TL），改變行4110的GOTO 50爲RETURN指令及在行160的末端加上：GOSUB 4000便成。

每當程序需要讀出指引文句時，用者祇須在適當的程式句加上MB $＝TL $（N）：CALL 16800：CALL 17280等指令群便可，而其中的N便是用者選定要讀出的文字句之參數。

同樣，你可進一步將SOUND UTILITY的HELLO 程式加上“GREETINGS”的聲音（意指歡迎）。甚至，用者可先自行設計了幾個不同聲音的HELLO程式，（如GREE—TINGS 1、GREETINGS 2、GREETINGS 3 至GREE—TINGS N 個等），在讀入SOUND UTILITY 程式前便可隨機選擇其一，那麽整個程式便更趨完美化，可謂有「聲」有色呢！（發聲的HELLO程式，可參閱程式2和3）

最後，還要提醒一些電腦初哥，若你們將MOCKING BOARD的文字檔案搬至其他磁碟上，這必須緊記同時要將SPEECH WRITER和4個文字與聲率解碼程式（SOUND UTILITY 中的機械碼副程式）存放於同一張的資料磁碟上。因爲主程式與資料檔案存於一磁碟上，是可避免插錯資料

```
1) INPUT NEW TEXT
2) SPEAK TEXT
3) SAVE TEXT TO  DISK
4) LOAD TEXT FROM DISK
5) MODIFY A LINE
6) INSERT LINE(S)
7) APPEND LINE(S)
8) DELETE LINE(S)
9) QUIT
```

磁碟的危險，而且操作也更爲方便。

至於4個文字與聲率解碼程式的拷貝方法，用者祇須先執行120—150 的程式行，繼而鍵入BSAVE SPEECH ALGORITHM，A $4009，L $1F80 便可。但相對以BA—SIC 編寫的SPEECH WRITER便要將原有的行120 —150刪去，取而代之便是120 PRINT D $ “BLOAD SPEE—CH ALGORITHM ”。

其實，MOCKINGBOARD 還有千千萬萬種的發聲功能，這是有待我們去發掘和利用。若所有的教育或商用程式均能善用MOCKINGBOARD的功能，那麽人類對電腦的恐懼感便可日益消減了！

程式 1

```
0  REM  **  SPEECH WRITER  **
1  REM  **  BY PROF. WONG  **
2  REM  **  COPYRIGHT (C)  **
3  REM  **       1985      **
4  LOMEM: 24448: PRINT  CHR$ (4);
     "MAXFILES1"
5 ML = 400: REM  MAXIMUM # OF LIN
     ES
10 D$ =  CHR$ (4): DIM TL$(ML): GOSUB
     50
20 LO% = SL%:HO% = SH%: REM  "CLE
     AN SLATE" POINTERS
48  GOTO 120
49  REM  SELECTIVE STRING PRESERV
     ATION
50 SL% =  PEEK (111):SH% =  PEEK
     (112): RETURN
51  POKE 111,SL%: POKE 112,SH%: RETURN

59  REM  BAD ENTRY MESSAGE
60  INVERSE : PRINT " BAD LINE NU
     MBER "; CHR$ (7): NORMAL : RETURN

99  REM  LOAD TEXT-TO-SPEECH ALGO
     RITHM
120  PRINT D$;"BLOADT-S2.OBJ"
130  PRINT D$;"BLOADRULES2.OBJ"
140  PRINT D$;"BLOADRNDX2.OBJ"
150  PRINT D$;"BLOADPHIDRV2.OBJ"
160  POKE 1022,161: POKE 1023,67:
      POKE 25,0
199  REM  MENU
200  TEXT : HOME : PRINT "1) INPU
     T NEW TEXT (CLEAN SLATE)"
210  PRINT : PRINT "2) SPEAK TEXT
     "
220  PRINT : PRINT "3) SAVE TEXT
     TO DISK "
230  PRINT : PRINT "4) LOAD TEXT
     FROM DISK "
240  PRINT : PRINT "5) MODIFY A L
     INE "
250  PRINT : PRINT "6) INSERT LIN
     E(S) "
260  PRINT : PRINT "7) APPEND LIN
     E(S) "
270  PRINT : PRINT "8) DELETE LIN
     E(S) "
280  PRINT : PRINT "9) QUIT "
300  PRINT : PRINT : INPUT "-> WH
     ICH ? ";IN$
310 IN% =  VAL (IN$)
320  IF IN% < 1 OR IN% > 9 THEN 2
     00
330  ON IN% GOSUB 1000,2000,3000,
     4000,5000,6000,7000,8000,600
     00
340  GOTO 200
999  REM  CLEAN SLATE AND INPUT N
     EW TEXT
1000 LN = 0:TL = 0
1010 SL% = LO%:SH% = HO%: GOSUB 5
     1
1020  TEXT : HOME : VTAB 3: PRINT
     "MAXIMUM LINE LENGTH = 256 C
     HARACTERS    INPUT ! TO QUIT
     ": POKE 34,5
1100  GOSUB 50: HOME : PRINT TL$(
     LN):LN = LN + 1: PRINT
1110 VT =  PEEK (37) + 1
1120  PRINT LN;: INPUT ") ";MB$: IF
     MB$ = "" THEN  VTAB VT: GOTO
     1120
1130  IF MB$ = "!" THEN TL = LN -
     1: RETURN
1140  CALL 16800: CALL 17280: REM
     SPEAK LINE
1150  CALL  - 958: PRINT : PRINT
```

選購電腦須知

市面上各類型電腦品種繁多，令人眼花繚亂，一不小心，
購買到不合用之電腦，除浪費金錢外，更費時失事。
下列各點，希望能幫助各位選購合適的電腦。

甲／硬件

- 必須大廠製造，質量方有保證。
- 強大擴展功能，以備日後擴展用途。

乙／軟件

- 要種類繁多，能適合多方面需要。
- 經常推陳出新，力求完善。

丙／操作

- 操作簡便，初學者感到易學易用。
- 同時具有複雜功能，令資深用家亦有發揮餘地。

丁／其他

- 主機及配件之價格要相宜。
- 有保用期，維修服務快捷，取價合理。



註：目前美國確達64型 (Commodore 64) 電腦全部符合上述要求，如欲取得詳細資料，請與下列商號聯絡。

香港：
- 正一電腦公司　5－504591
- 怡通電業　5－7956249
- Microage Superstore　5－738282
- RTC Ltd.　5－258331

九龍：
- 樂聲電腦電業公司　3－680878
- 威發玩具　3－313218
- 金獅電腦公司　3－660645
- 天龍電腦公司　3－7235698
- 威能電腦有限公司　3－309225
- VIC Trading Co.　3－7297021
- Multi M Co. Ltd.　3－7236201
- 東亞公司　3－7234392
- N-BIT Computer Supplies　3－7233382

新界：
- 屯門電腦中心　0－830486

COMMODORE 64

總代理：鷹達企業有限公司

香港灣仔洛克道194－200號
東新商業中心17樓
電話：5－8934447（六線）

令蘋果發言的最佳工具 SPEECH WRITER

```
     "DOES THAT SOUND ACCEPTABLE?
      ";
1160  GET IN$: IF IN$ <  > "Y" AND
     IN$ <  > "N" THEN 1160
1170  PRINT IN$: IF IN$ = "N" THEN
      VTAB VT: CALL  - 958: PRINT
     LN;") ";MB$: VTAB VT: GOTO 1
     120
1180  GOSUB 51:TL$(LN) = "" + MB$
     : GOTO 1100
1999  REM  SPEAK TEXT IN RAM
2000  HOME : VTAB 4: PRINT "THERE
      ARE ";TL;" LINES OF TEXT IN
      MEMORY. "
2010  VTAB 7: PRINT "1) SPEAK ALL
      LINES"
2020  PRINT : PRINT "2) SPEAK A C
     HOSEN RANGE OF LINES"
2030  VTAB 14: INPUT "-> WHICH ?
      ";IN$: IF IN$ = "" THEN  RETURN

2040 IN% =  VAL (IN$)
2050  IF IN% < 1 OR  > 2 THEN 200
     0
2060  IF IN% = 1 THEN L1 = 1:L2 =
     TL: GOTO 2200
2100  VTAB 16: INPUT "FIRST LINE
     NUMBER = ";IN$
2110 IN% =  VAL (IN$): IF IN% < 1
      OR IN% > TL THEN  GOSUB 60:
      GOTO 2100
2120 L1 = IN%: CALL  - 958
2130  VTAB 18: INPUT "LAST LINE N
     UMBER = ";IN$
2140 IN% =  VAL (IN$): IF IN% < L
     1 OR IN% > TL THEN  GOSUB 60
     : GOTO 2130
2150 L2 = IN%
2200  HOME : FOR LN = L1 TO L2
2210  GOSUB 51
2220 MB$ = TL$(LN): PRINT LN;") "
     ;MB$
2230  CALL 16800: CALL 17280
2240  ON  PEEK (25) = 255 GOTO 22
     40: NEXT : RETURN
2999  REM  SAVE TEXT TO DISK
3000  HOME : VTAB 4: PRINT "PLEAS
     E ENTER A NAME FOR THE FILE:
     "
3010  PRINT : INPUT N$: IF N$ = "
     " THEN  RETURN
3020  IF  LEFT$ (N$,5) <  > "MOCK
     ." THEN N$ = "MOCK." + N$
3030  PRINT D$;"OPEN";N$
3040  PRINT D$;"WRITE";N$
3050  GOSUB 51: PRINT TL
3060  FOR LN = 1 TO TL
3070  PRINT TL$(LN)
3080  NEXT
3090  PRINT D$;"CLOSE"
3100  RETURN
3999  REM  LOAD TEXT FROM DISK
4000  HOME : VTAB 4: INPUT "INSER
     T DISK CONTAINING THE FILE A
     ND PRESS RETURN";IN$
4010  PRINT D$;"CATALOG": PRINT
4020  INPUT "NAME OF FILE: ";N$: IF
     N$ = "" THEN  RETURN
4030  IF  LEFT$ (N$,5) <  > "MOCK
     ." THEN N$ = "MOCK." + N$
4040  PRINT D$;"OPEN";N$
4050  PRINT D$;"READ";N$
4060 SL% = LO%:SH% = HO%: GOSUB 5
     1
4070  INPUT TL
4080  FOR LN = 1 TO TL
4090  INPUT TL$(LN)
4100  NEXT
4110  PRINT D$;"CLOSE": GOTO 50
4999  REM  MODIFY AN EXISTING LIN
     E
5000  HOME
5005  VTAB 10: PRINT "THERE ARE "
     ;TL;" LINES IN MEMORY."
5010  INPUT "WHICH LINE DO YOU WA
     NT TO CHANGE? ";IN$: IF IN$ =
     "" THEN  RETURN
5020 IN% =  VAL (IN$): IF IN% < 1
      OR IN% > TL THEN  GOSUB 60:
      GOTO 5005
5030 LN = IN%: CALL  - 958
5040  HOME : ON LN = 1 GOTO 5050:
      PRINT LN - 1;") ";TL$(LN -
      1): PRINT
5050 VT =  PEEK (37) + 1: PRINT L
     N;") ";TL$(LN): PRINT
5060  ON LN = TL GOTO 5070: PRINT
     LN + 1;") ";TL$(LN + 1)
5070  VTAB VT: PRINT LN;: INPUT "
     ) ";MB$
5080  CALL 16800: CALL 17280: REM
               SPEAK LINE
5090  CALL  - 958: PRINT : PRINT
     "DOES THAT SOUND ACCEPTABLE?
      ";
5100  GET IN$: IF IN$ <  > "Y" AND
     IN$ <  > "N" THEN 5100
5110  PRINT IN$: IF IN$ = "N" THEN
      VTAB VT: CALL  - 958: PRINT
     LN;") ";MB$: GOTO 5070
5120  GOSUB 51:TL$(LN) = "" + MB$
     : GOSUB 50: RETURN
5999  REM  INSERT LINE(S) BETWEEN
      EXISTING LINES
6000  HOME : VTAB 4: PRINT "THERE
      ARE ";TL;" LINES IN THE FIL
     E."
6010  VTAB 10: INPUT "INSERT HOW
     MANY LINES? ";IN$: IF IN$ =
     "" THEN  RETURN
6020 IN% =  VAL (IN$): IF IN% < 1
      OR IN% + TL > ML THEN  GOSUB
     60: GOTO 6010
6030 IL = IN%: CALL  - 958
6040  VTAB 12: INPUT "INSERT AFTE
     R LINE #";IN$
```

令蘋果發言的最佳工具 SPEECH WRITER

```
6050 IN% =  VAL (IN$): IF IN% < 0
 OR IN% > TL THEN  GOSUB 60:
 GOTO 6040
6060 L1 = IN%: CALL  - 958: GOSUB
 51
6070  FOR LN = TL TO L1 + 1 STEP
  - 1
6080 TL$(LN + IL) = TL$(LN)
6090  NEXT :TL = TL + IL: GOSUB 5
 0
6100  FOR LN = L1 + 1 TO L1 + IL
6110  HOME : PRINT TL$(LN - 1): PRINT

6120 VT =  PEEK (37) + 1: PRINT L
 N;
6130  INPUT ") ";MB$
6140  CALL 16800: CALL 17280: REM
 SPEAK LINE
6150  CALL  - 958: PRINT : PRINT
 "DOES THAT SOUND ACCEPTABLE?
 ";
6160  GET IN$: IF IN$ <  > "Y" AND
 IN$ <  > "N" THEN 6160
6170  PRINT IN$: IF IN$ <  > "N" THEN
  VTAB VT: CALL  - 958: PRINT
 LN;") ";MB$: VTAB VT: GOTO 6
 120
6180  GOSUB 51:TL$(LN) = "" + MB$
 : GOSUB 50
6190  NEXT : RETURN
6999  REM  APPENDS LINE(S)
7000  IF TL = 0 THEN 1000
7010  GOSUB 51:LN = TL: GOTO 1020

7999  REM  DELETE LINES(S)
8000  HOME : VTAB 4: PRINT "THERE
 ARE ";TL;" LINES IN THE FIL
 E."
8010  VTAB 10: INPUT "FIRST LINE
 TO DELETE: #";IN$: IF IN$ =
 "" THEN  RETURN
8020 IN% =  VAL (IN$): IF IN% < 1
 OR IN% > TL THEN  GOSUB 60:
 GOTO 8010
8030 L1 = IN%: CALL  - 958
8040  VTAB 12: INPUT "LAST LINE T
 O DELETE: #";IN$: IF IN$ = "
 " THEN  RETURN
8050 IN% =  VAL (IN$): IF IN% < L
 1 OR IN% > TL THEN  GOSUB 60
 : GOTO 8040
8060 L2 = IN%:DL = L2 - L1 + 1:TL
  = TL - DL: CALL  - 958
8070  GOSUB 51: FOR LN = L1 TO TL

8080 TL$(LN) = TL$(LN + DL)
8090  NEXT : GOTO 50
59999  REM  QUIT
60000  POP : HOME : END
```

程式 2

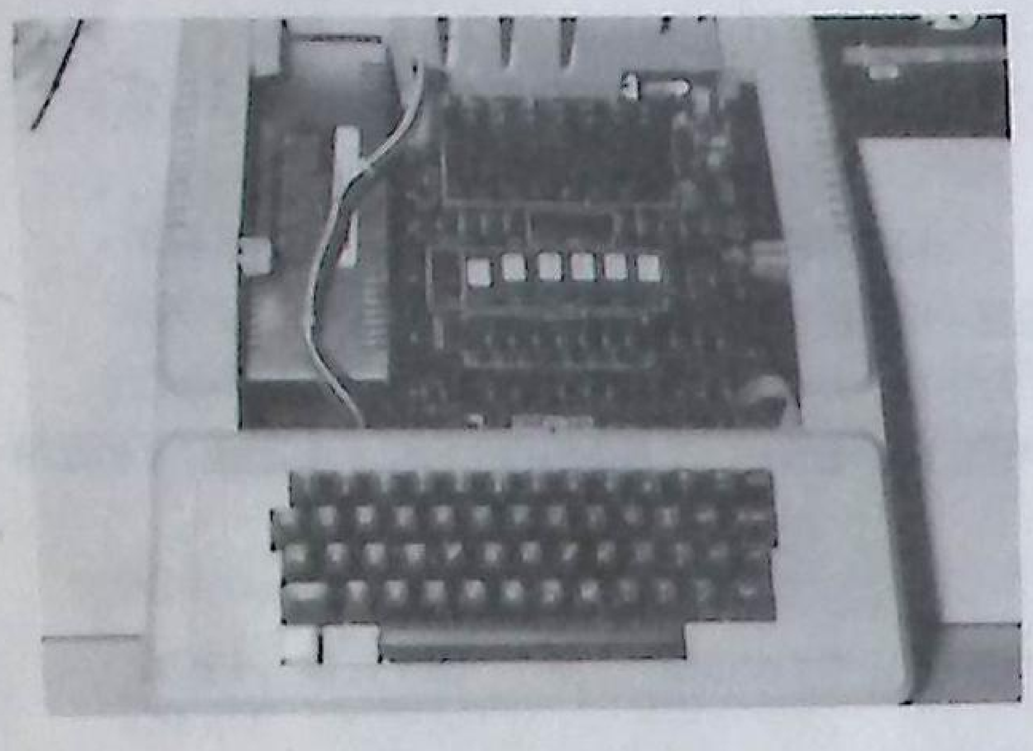
```
20 N = 10: REM  10 GREETINGS FILE
 S
1000  TEXT : HOME : VTAB 8: PRINT
 "PLEASE PRESS A KEY."
1010 X% =  RND (0) * N + 1
1020  IF  PEEK ( - 16384) < 128 THEN
 1010
1030  POKE  - 16368,0
4000 N$ = "GREETINGS" +  STR$ (X%
 )
```

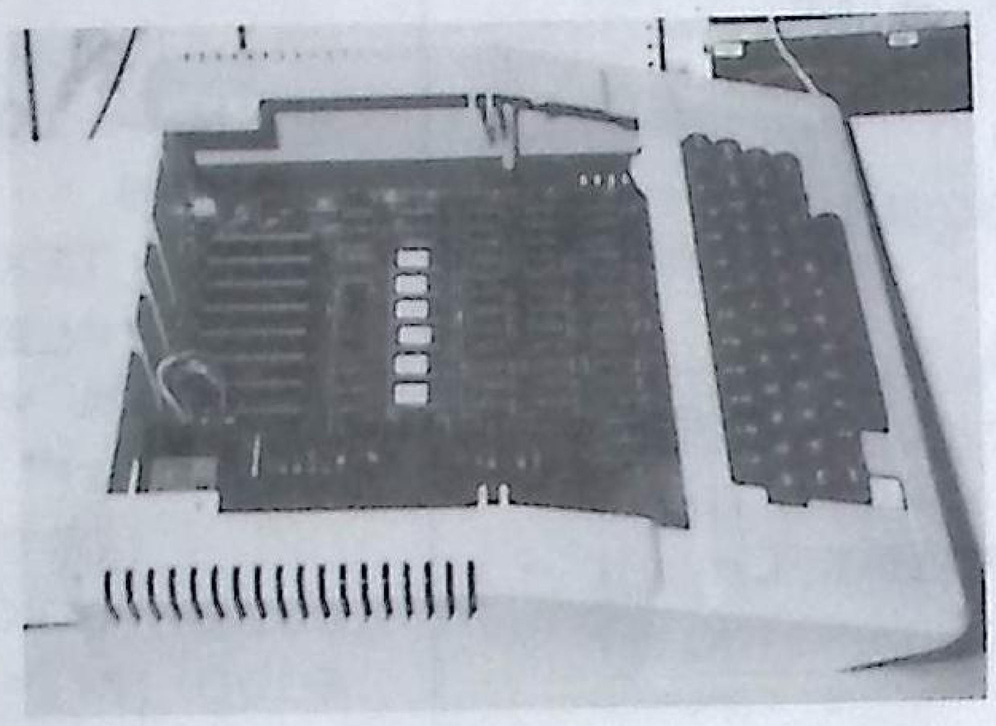
程式 3

```
0  REM  "TALKING HELLO"
10 D$ =  CHR$ (4)
99  REM  LOAD TEXT-TO-SPEECH ALGO
 RITHM
120  PRINT D$;"BLOADT-S2.OBJ"
130  PRINT D$;"BLOADRULES2.OBJ"
140  PRINT D$;"BLOADRNDX2.OBJ"
150  PRINT D$;"BLOADPHIDRV2.OBJ"
160  POKE 1022,161: POKE 1023,67:
 POKE 25,0
2000  TEXT : HOME : PRINT D$;"CAT
 ALOG": REM  CATALOG COMMAND
 MAY BE DROPPED IF NOT APPROP
 RIATE FOR YOUR APPLICATION.
2200  GOSUB 4000:L1 = 1:L2 = TL
2210  FOR LN = L1 TO L2
2220 MB$ = TL$(LN)
2230  CALL 16800: CALL 17280
2240  ON  PEEK (25) = 255 GOTO 22
 40: NEXT
2999  REM  END PROGRAM OR RUN USE
 R PROGRAM
3000  END
3999  REM  LOAD TEXT FROM DISK
4000 N$ = "GREETING"
4030  IF  LEFT$ (N$,5) <  > "MOCK
 ." THEN N$ = "MOCK." + N$
4040  PRINT D$;"OPEN";N$
4050  PRINT D$;"READ";N$
4070  INPUT TL
4075  DIM TL$(TL)
4080  FOR LN = 1 TO TL
4090  INPUT TL$(LN)
4100  NEXT
4110  PRINT D$;"CLOSE": RETURN
```

如發覺自己的電腦出現問題？要立刻修理，否則，會造成嚴重損壞或傷害！但本中心對你的電腦極為關心，特設維修服務站處理有關電腦問題，有專業人仕解答或修理，如有詢問，請打熱線電話：3- 7783923

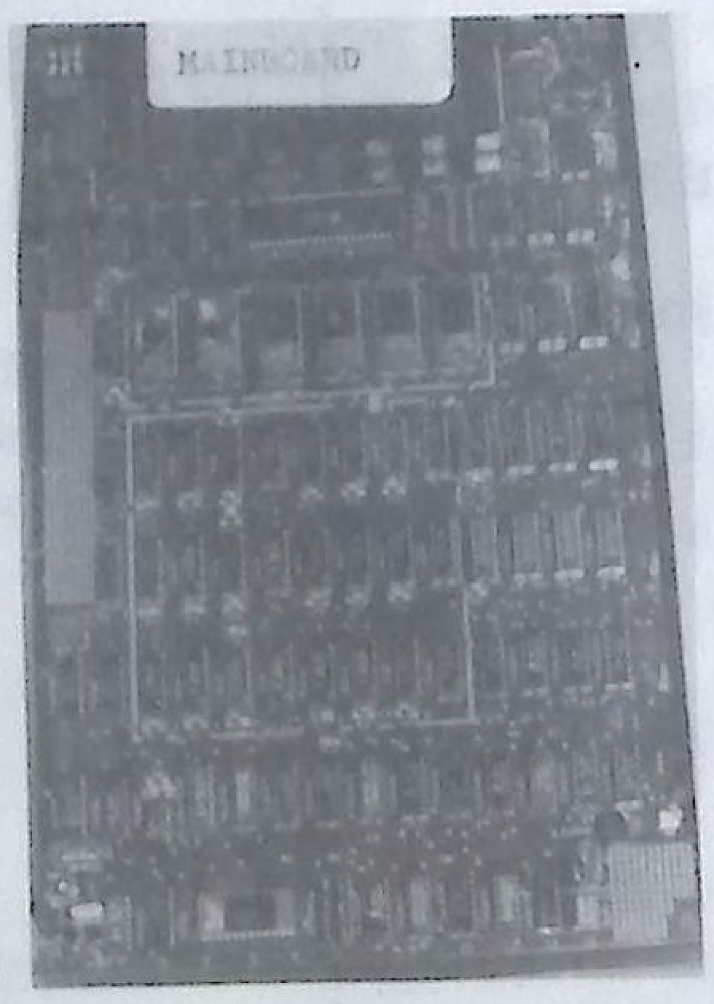
＄價錢低廉＄





〈專營〉

多款 電腦硬件，軟件，配件，零件，I.C. 盒帶，磁碟，書籍，保養及維修。



—批發—

大量電腦日本48K主板，咭件及散料，適合行家及回國外地商戶，低價出售，平靚，有意請來電查詢。

•收購•

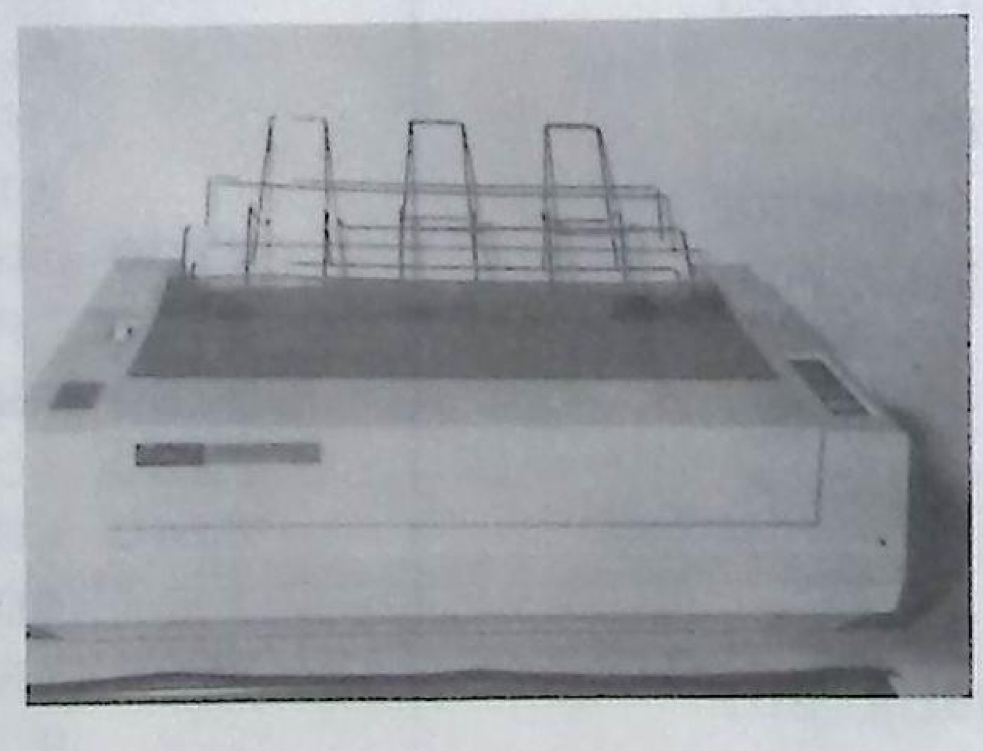
二手電腦，任何牌子，磁碟機，顯示器，及其它電腦附件一律歡迎，全港最高，不妨試試。

獨家發售

街機（粗暴型）

JOYSTICK，遊戲機中心，一樣刺激緊張，內設强力閃燈，可裝上內藏揚聲器。

只售：199.00





123 電腦中心 COMPUTER CENTRE

電腦維修保養服務站

COMPUTER MAINTENANCE SERVICE

地址：深水埗南昌街白田購物中心一樓 102 室（南昌戲院後面）

電話：3-7783923

高解像圖魔術師

利用本程式，可以玩弄許多高解像圖花巧，仿如一個魔術師。

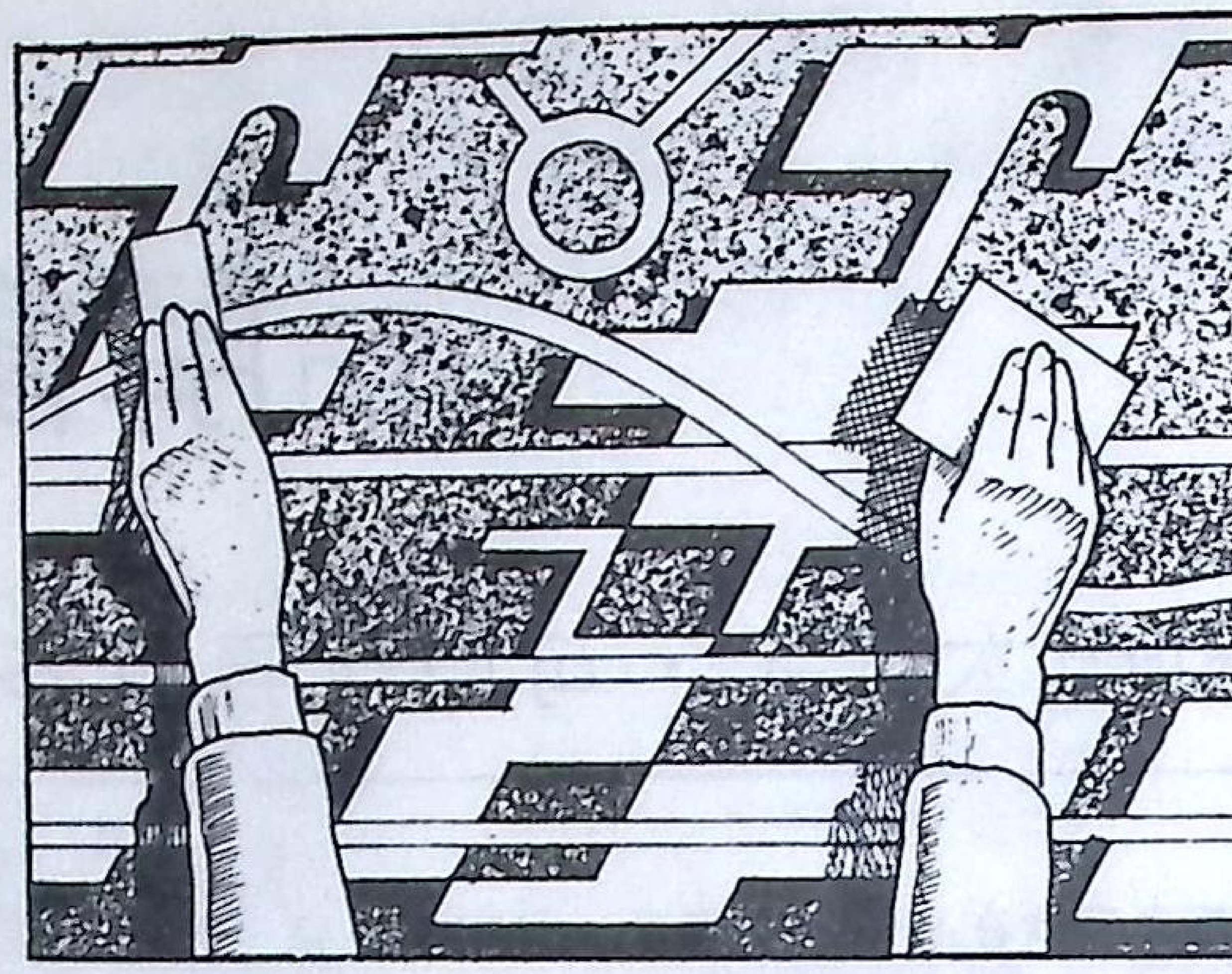
大家都會喜歡在高解像頁裏弄一些魔術玩兒，那麽千萬不要錯過本文的一個用機械碼寫成的程式。

這個魔術程式能做到的效果會令你覺得詫異，他們包括：

• 輕而易擧將高解像頁上、下、左、右的移動。如此，可以讓你任意將高解像圖形移到你所要的位置，方便加插其他圖形，或將圖形的色彩改變。
• 將高解像圖形以反相顯示
• 將高解像螢幕BYTE的顏色BIT 改變爲其他，方便觀看其他效果

```
10  REM **************************
20  REM *  HI-RES MAGIC DRIVER *
30  REM *  BY COMPUHOUSE       *
40  REM *  COPYRIGHT (C) 1984  *
50  REM *  BY COMPUTING AGE    *
60  REM *                      *
70  REM **************************
80 D$ =  CHR$ (4)
90  ONERR  GOTO 360
100  HOME : VTAB 2: HTAB 10: INVERSE
    : PRINT "HI-RES MAGIC DRIVER
    ": NORMAL
110  VTAB 12: CALL  - 958: VTAB 2
    2: PRINT "'?' FOR DISK CATAL
    OG": PRINT "<RETURN> TO SKIP
    ": VTAB 12: PRINT "FILE NAME
     FOR PAGE 1 PICTURE:": INPUT
    "";P1$
120  IF P1$ = "?" THEN  HOME : PRINT
    D$"CATALOG": PRINT "PRESS AN
    Y KEY TO CONTINUE": GET K$: PRINT
    : HOME : GOTO 110
130  IF P1$ = "" THEN 150
140  PRINT D$;"BLOAD";P1$;",A$200
    0"
150  VTAB 12: CALL  - 958: VTAB 2
    2: PRINT "'?' FOR DISK CATAL
    OG": PRINT "<RETURN> TO SKIP
    ": VTAB 12: PRINT "FILE NAME
     FOR PAGE 2 PICTURE:": INPUT
    "";P2$
160  IF P2$ = "?" THEN  HOME : PRINT
    D$"CATALOG": PRINT "PRESS AN
    Y KEY TO CONTINUE": GET K$: PRINT
    : HOME : GOTO 150
170  IF P2$ = "" THEN 190
180  PRINT D$;"BLOAD";P2$;",A$400
    0"
190  PRINT D$;"BLOAD HI.RES.HOUDI
    NI"
200  CALL 36608: TEXT : HOME
210  HOME : VTAB 12: PRINT "DO YO
    U REALLY WANT TO QUIT? (Y/N)
    ";: GET K$: PRINT : IF K$ =
    "N" THEN  CALL 36608: GOTO 2
    10
220  IF K$ <  > "Y" THEN 210
230  HOME : VTAB 12: PRINT "LOAD
    NEW PICTURES AND RE-START? (
    Y/N)";: GET K$: PRINT : IF K
    $ = "Y" THEN 100
240  IF K$ <  > "N" THEN 230
250  ONERR  GOTO 380
260  HOME : VTAB 12: PRINT "SAVE
    PICTURE ON PAGE 1? (Y/N)";: GET
    K$: PRINT : IF K$ = "N" THEN
    300
270  IF K$ <  > "Y" THEN 260
280  VTAB 12: CALL  - 958: PRINT
    "ENTER FILE NAME:": INPUT ""
    ;F$: IF  LEN (F$) > 15 OR  VAL
    (F$) > 0 THEN  PRINT "ILLEGA
    L FILE NAME. TRY AGAIN.": FOR
    I = 1 TO 1000: NEXT : GOTO 2
    80
290  PRINT D$"BSAVE";F$;",A$2000,
    L$2000"
300  HOME : VTAB 12: PRINT "SAVE
    PICTURE ON PAGE 2? (Y/N)";: GET
    K$: PRINT : IF K$ = "N" THEN
    340
310  IF K$ <  > "Y" THEN 300
320  VTAB 12: CALL  - 958: PRINT
    "ENTER FILE NAME:": INPUT ""
    ;F$: IF  LEN (F$) > 15 OR  VAL
    (F$) > 0 THEN  PRINT "ILLEGA
    L FILE NAME. TRY AGAIN.": FOR
    I = 1 TO 1000: NEXT : GOTO 3
    20
330  PRINT D$"BSAVE";F$;",A$4000,
    L$2000"
340  HOME : END
350  REM  ERROR TRAP #1
360  VTAB 22: PRINT "ERROR NUMBER
    "; PEEK (222): PRINT "PRESS
    ANY KEY TO START AGAIN";: GET
    K$: GOTO 100
370  REM  ERROR TRAP #2
380  VTAB 22: PRINT "ERROR NUMBER
    "; PEEK (222): PRINT "PRESS
    ANY KEY TO TRY AGAIN";: GET
    K$: GOTO 260
```

高解像圖魔術師



• 隨意調換第一及第二頁高解像圖
• 將兩頁高解像合二爲一
• 將一個圖形從一頁中的位置再COPY 到另一位置

這個魔術師程式在控制方面非常簡易，執行程式後，螢幕下半部便會出現一系列的指令，他們每個的解釋如下：

（→）：以一個BYTE爲單位的速度將螢幕右移。螢幕上所有的東西都會同時移動，原先在最右邊的東西會在螢幕左邊重新出現。由於單數目行的點會向雙數目行移動，因此會改變了其色彩，用彩色電視的讀者自會清楚其分辨。

（←）：將螢幕以1 BYTE 的速度向左移，情形如上述（→）鍵所做的剛巧相反。

（>或・）：將螢幕以1 BIT （注意是1 BIT而非1BYTE）的速度向右移，換言之，是以一點或一像素的速度移動。由於單數目行會變爲雙數目行，所以移動時色彩是會有改變。

（<或，）：和上述>或・指令相反，它將螢幕以1BIT 的速度向左移。

（A）：將螢幕上所有的東西向上移。在最上邊的東西會從新於螢幕下邊出現。由於所有點所位於的行列沒有改變，故不會有任何色彩的轉換。

（Z）：和A鍵的功能相反，將螢幕向下捲。

（I）：將高解像圖的顏色反相。這個指令將所有被設定的BIT取消，而將未有設定的BIT加上顏色。例如黑白互換、綠藍對調等。

（C）：改變BIT的顏色。這個指令將所有被設定的COL－OR BIT（即每個BYTE中的第七個BIT）取消，並將其他未被設定的COLOR BIT 設定。它將綠紅、紫藍的顏色對調。或可將WHITE 1轉爲WHITE 2及BLACK1 轉爲BLACK2 。

（F）：觀看全高解像或高解像圖文字混合圖。當執行本程式時，螢幕是用高解像圖及文字混合模式顯示，按F F鍵後，下列四行的文字指令說明會消失，再按F，文字會出現。這指令對於觀看第二頁高解像圖沒有效，因爲第二頁高解像頁是沒有文字混合的。

（M）：將兩頁圖像湊合成一頁。假如你是在觀看第一頁圖像，而按動M，則第二頁會和第一頁的圖像湊合而成一幅新的圖像，但第二頁的東西不會有改變。相反，假如所觀看的是第二頁圖像，按下M後，兩圖頁會合併，但第一頁的東西不會受影响。假如認爲合併效果不理想，可再按M回復原來面目。

（CTRL @）：清理高解像頁。

（F）：隨意對調第一或第二頁。

（Q）：離開本程式。要再進入本程式，可CALL 36608。

爲了方便，我另外編寫了一個APPLESOFT 的驅動程式，方便讀者使用，各位只要RUN 這驅動程式便可。至於本程式的用途，則要視乎你本身的想像力。

各位可利用他來製造一些奇異效果的畫面，擁有本期程式磁碟的讀者，大可利用版頭來作一個開始。

本文共有兩個程式。有關機械碼的輸入方法，各位可參閱本刊第三期。

本文的主程式僅爲1.6K BYTE 長，它的原程式列出如後，基本上它可以加接到其它高解像程式之後或其它應用上，這一點留待給各位好了。

主程式是採用BIG MAC 組合器編寫而成，由於它擁有MACRO功能，因此可以把經常叫用的程式使用簡化，只要以>>>符號指明。若果沒有這個組合器的朋友請勿鍵入此原程式，只要鍵入機械碼即可。對於擁有BIG MAC MAC－RO 組合器的朋友在鍵入本程式時對>>>這個MACRO符號應該只鍵入其相同行號的爲首一行。以後有機會我會介紹MACRO組合器的使用。

主程式中第14至52行是定出一些位址作爲變數或副程式的進入點。第54至82行是MACRO所在，第54行的D0 0 是告訴組合器不要對MACRO組合，第82行的FIN是FINISH的意思，即可以對此行以下的命令組合。

第85行開始程式首先選擇高解像圖形頁，然後選擇混合文字圖形模式將操作指令印出在最下面4行（第95行至114行），命令的輸入是由第117至151 行的程式執行，若果按入的命令是程式接納的話則會交由159行至170行分別跳往其副程式的進入點去。若果按下"A"鍵的話則跳到176行至215行執行，使螢幕所有位元都向上移，使產生上捲效果，相反按入"Z"鍵的話會交由218行至260行將螢幕向下捲動。

按入"→"鍵會使螢幕向右捲動，由272至359行執行。按入"←"鍵會使螢幕向左捲動，由361至452行執行。

使螢幕上圖形產生反相效果是由第455至486行執行，主要是將記憶內的位元做一個EXCLUSIVE OR （異門），再將結果存入其原來位置上。

第489至524行是給用家設定選擇模式之用，可以混合圖形／文字或全圖形顯示。第527至548行是將兩頁圖形合併成爲一頁，頁551至560行是用作清除圖形所用。

第561至608行是螢幕垂直座標的高低位元位址對照表，第611至647行是操作命令的HSC Ⅱ字元表。■

```
*****************************
*                           *
* HIGH RESOLUTION MAGIC     *
*                           *
*     BY  Computing Age     *
*                           *
*   ALL RIGHTS RESERVED     *
*                           *
*****************************

         ORG   $8F00      ;


FMFLAG   EQU   0          ;
COLORBIT EQU   1          ;
NEWBYTE  EQU   2          ;
HRPAGE   EQU   $19        ;
OLDPTR   EQU   $1A        ;
NEWPTR   EQU   $1C        ;
ROW      EQU   $1E        ;
COL      EQU   $1F        ;


RIT^ARROW=     $95        ;
LFT^ARROW=     $88        ;
UP^A     =     "A"        ;
DOWN^Z   =     "Z"        ;
CTRL^@   =     $80        ;

NUMCOL   =     39         ;
TOP^ROW  =     0          ;
BOT^ROW  =     191        ;


CH       EQU   $24        ;
HPAGE    EQU   $E6        ;
APLSOFT  EQU   $3D0       ;
KEYBD    EQU   $C000      ;
STROBE   EQU   $C010      ;
SHOW     EQU   $C050      ;
FULLSCRN EQU   $C052      ;
MXEDSCRN EQU   $C053      ;
FLIP1    EQU   $C054      ;
FLIP2    EQU   $C055      ;
HRSCRN   EQU   $C057      ;
HCLR     EQU   $F3F2      ;
SETTXT   EQU   $FB39      ;
TABV     EQU   $FB5B      ;
HOME     EQU   $FC58      ;
COUT     EQU   $FDED      ;
BELL     EQU   $FF3A      ;

***************************************************

         DO    0          ;

SETPTR   MAC              ;
         LDA   YLOW,X     ;
         STA   ]1         ;
         CLC              ;
         LDA   YHIGH,X    ;
         ADC   HRPAGE     ;
         STA   ]1+1       ;
         <<<              ;

TABXY    MAC              ;
         LDA   #]1        ;
         STA   CH         ;
         LDA   #]2        ;
         JSR   TABV       ;
         <<<              ;

MESSAGE  MAC              ;
         LDY   #0         ;
                75  MSGLOOP  LDA  ]1,Y         ;
                76           BEQ  MEND         ;
                77           JSR  COUT         ;
                78           INY               ;
                79           BNE  MSGLOOP      ;
                80  MEND     <<<               ;
                81
                82           FIN               ;
                83
                84
8F00: 2C 57 C0  85           BIT  HRSCRN       ;
8F03: 2C 54 C0  86           BIT  FLIP1        ;
8F06: 2C 53 C0  87           BIT  MXEDSCRN     ;
8F09: 2C 50 C0  88           BIT  SHOW         ;
8F0C: 2C 10 C0  89           BIT  STROBE       ;
8F0F: A9 00     90           LDA  #0           ;
8F11: 85 19     91           STA  HRPAGE       ;
8F13: 85 00     92           STA  FMFLAG       ;
                93
                94
8F15: 20 58 FC  95           JSR  HOME         ;
                96           >>>  TABXY.0;20
8F18: A9 00     96           LDA  #0           ;
8F1A: 85 24     96           STA  CH           ;
8F1C: A9 14     96           LDA  #20          ;
8F1E: 20 5B FB  96           JSR  TABV         ;
                96           <<<               ;
                97           >>>  MESSAGE.SCLCMNDS
8F21: A0 00     97           LDY  #0           ;
8F23: B9 D6 94  97  MSGLOOP  LDA  SCLCMNDS,Y   ;
8F26: F0 06     97           BEQ  MEND         ;
8F28: 20 ED FD  97           JSR  COUT         ;
8F2B: C8        97           INY               ;
8F2C: D0 F5     97           BNE  MSGLOOP      ;
                97  MEND     <<<               ;
                98           >>>  TABXY.0;21
8F2E: A9 00     98           LDA  #0           ;
8F30: 85 24     98           STA  CH           ;
8F32: A9 15     98           LDA  #21          ;
8F34: 20 5B FB  98           JSR  TABV         ;
                98           <<<               ;
                99           >>>  MESSAGE.INV:I
8F37: A0 00     99           LDY  #0           ;
8F39: B9 FA 94  99  MSGLOOP  LDA  INV:I,Y      ;
8F3C: F0 06     99           BEQ  MEND         ;
8F3E: 20 ED FD  99           JSR  COUT         ;
8F41: C8        99           INY               ;
8F42: D0 F5     99           BNE  MSGLOOP      ;
                99  MEND     <<<               ;
               100           >>>  TABXY.20;21
8F44: A9 14    100           LDA  #20          ;
8F46: 85 24    100           STA  CH           ;
8F48: A9 15    100           LDA  #21          ;
8F4A: 20 5B FB 100           JSR  TABV         ;
               100           <<<               ;
               101           >>>  MESSAGE.CHGBIT:C
8F4D: A0 00    101           LDY  #0           ;
8F4F: B9 0B 95 101  MSGLOOP  LDA  CHGBIT:C,Y   ;
8F52: F0 06    101           BEQ  MEND         ;
8F54: 20 ED FD 101           JSR  COUT         ;
8F57: C8       101           INY               ;
8F58: D0 F5    101           BNE  MSGLOOP      ;
               101  MEND     <<<               ;
               102           >>>  TABXY.0;22
8F5A: A9 00    102           LDA  #0           ;
8F5C: 85 24    102           STA  CH           ;
8F5E: A9 16    102           LDA  #22          ;
8F60: 20 5B FB 102           JSR  TABV         ;
               102           <<<               ;
               103           >>>  MESSAGE.FLMX:F
8F63: A0 00    103           LDY  #0           ;
8F65: B9 1E 95 103  MSGLOOP  LDA  FLMX:F,Y     ;
8F68: F0 06    103           BEQ  MEND         ;
8F6A: 20 ED FD 103           JSR  COUT         ;
```

```
8F6D: C8         103          INY              ;
8F6E: D0 F5      103          BNE  MSGLOOP     ;
                 103   MEND   <<<              ;
                 104          >>>  TABXY.20;22
8F70: A9 14      104          LDA  #20         ;
8F72: 85 24      104          STA  CH          ;
8F74: A9 16      104          LDA  #22         ;
8F76: 20 5B FB   104          JSR  TABV        ;
                 104          <<<              ;
                 105          >>>  MESSAGE.MERGE:M
8F79: A0 00      105          LDY  #0          ;
8F7B: B9 32 95   105   MSGLOOP LDA MERGE:M,Y   ;
8F7E: F0 06      105          BEQ  MEND        ;
8F80: 20 ED FD   105          JSR  COUT        ;
8F83: C8         105          INY              ;
8F84: D0 F5      105          BNE  MSGLOOP     ;
                 105   MEND   <<<              ;
                 106          >>>  TABXY.0;23
8F86: A9 00      106          LDA  #0          ;
8F88: 85 24      106          STA  CH          ;
8F8A: A9 17      106          LDA  #23         ;
8F8C: 20 5B FB   106          JSR  TABV        ;
                 106          <<<              ;
                 107          >>>  MESSAGE.CLEAR:@
8F8F: A0 00      107          LDY  #0          ;
8F91: B9 42 95   107   MSGLOOP LDA CLEAR:@,Y   ;
8F94: F0 06      107          BEQ  MEND        ;
8F96: 20 ED FD   107          JSR  COUT        ;
8F99: C8         107          INY              ;
8F9A: D0 F5      107          BNE  MSGLOOP     ;
                 107   MEND   <<<              ;
                 108          >>>  TABXY.5;23
8F9C: A9 05      108          LDA  #5          ;
8F9E: 85 24      108          STA  CH          ;
8FA0: A9 17      108          LDA  #23         ;
8FA2: 20 5B FB   108          JSR  TABV        ;
                 108          <<<              ;
8FA5: A9 00      109          LDA  #$0         ;
8FA7: 20 ED FD   110          JSR  COUT        ;
                 111          >>>  TABXY.20;23
8FAA: A9 14      111          LDA  #20         ;
8FAC: 85 24      111          STA  CH          ;
8FAE: A9 17      111          LDA  #23         ;
8FB0: 20 5B FB   111          JSR  TABV        ;
                 111          <<<              ;
                 112          >>>  MESSAGE.PAGE:P
8FB3: A0 00      112          LDY  #0          ;
8FB5: B9 56 95   112   MSGLOOP LDA PAGE:P,Y    ;
8FB8: F0 06      112          BEQ  MEND        ;
8FBA: 20 ED FD   112          JSR  COUT        ;
8FBD: C8         112          INY              ;
8FBE: D0 F5      112          BNE  MSGLOOP     ;
                 112   MEND   <<<              ;
                 113          >>>  TABXY.33;23
8FC0: A9 21      113          LDA  #33         ;
8FC2: 85 24      113          STA  CH          ;
8FC4: A9 17      113          LDA  #23         ;
8FC6: 20 5B FB   113          JSR  TABV        ;
                 113          <<<              ;
                 114          >>>  MESSAGE.QUIT:Q
8FC9: A0 00      114          LDY  #0          ;
8FCB: B9 62 95   114   MSGLOOP LDA QUIT:Q,Y    ;
8FCE: F0 06      114          BEQ  MEND        ;
8FD0: 20 ED FD   114          JSR  COUT        ;
8FD3: C8         114          INY              ;
8FD4: D0 F5      114          BNE  MSGLOOP     ;
                 114   MEND   <<<              ;
                 115
                 116
8FD6: AD 00 C0   117   KEYIN  LDA  KEYBD       ;
8FD9: 10 FB      118          BPL  KEYIN       ;
8FDB: 2C 10 C0   119          BIT  STROBE      ;
8FDE: C9 D1      120          CMP  #"Q"        ;
8FE0: F0 3E      121          BEQ  QUIT        ;
```

```
8FE2: C9 C1      122          CMP  #UP^A       ;
8FE4: F0 44      123          BEQ  SCRLUP      ;
8FE6: C9 DA      124          CMP  #DOWN^Z     ;
8FE8: F0 43      125          BEQ  SCRLDWN     ;
8FEA: C9 95      126          CMP  #RIT^ARROW  ;
8FEC: F0 42      127          BEQ  SCRLRIT     ;
8FEE: C9 88      128          CMP  #LFT^ARROW  ;
8FF0: F0 41      129          BEQ  SCRLLFT     ;
8FF2: C9 BE      130          CMP  #">"        ;
8FF4: F0 40      131          BEQ  MOVERIGHT   ;
8FF6: C9 AE      132          CMP  #"."        ;
8FF8: F0 3C      133          BEQ  MOVERIGHT   ;
8FFA: C9 BC      134          CMP  #"<"        ;
8FFC: F0 3B      135          BEQ  MOVELEFT    ;
8FFE: C9 AC      136          CMP  #","        ;
9000: F0 37      137          BEQ  MOVELEFT    ;
9002: C9 C9      138          CMP  #"I"        ;
9004: F0 36      139          BEQ  INVERSE     ;
9006: C9 C3      140          CMP  #"C"        ;
9008: F0 35      141          BEQ  CHHCLBIT    ;
900A: C9 D0      142          CMP  #"P"        ;
900C: F0 34      143          BEQ  PAGECHNG    ;
900E: C9 C6      144          CMP  #"F"        ;
9010: F0 33      145          BEQ  FULLMXD     ;
9012: C9 CD      146          CMP  #"M"        ;
9014: F0 32      147          BEQ  PICMERGE    ;
9016: C9 80      148          CMP  #CTRL^@     ;
9018: F0 31      149          BEQ  CLEARSCRN   ;
901A: 20 3A FF   150          JSR  BELL        ;
901D: 4C D6 8F   151          JMP  KEYIN       ;
                 152
9020: 2C 54 C0   153   QUIT   BIT  FLIP1       ;
9023: 20 39 FB   154          JSR  SETTXT      ;
9026: 20 58 FC   155          JSR  HOME        ;
9029: 60         156          RTS              ;
                 157
                 158
902A: 4C 4E 90   159   SCRLUP    JMP  SCROLL^UP       ;
902D: 4C CE 90   160   SCRLDWN   JMP  SCROLL^DOWN     ;
9030: 4C 58 91   161   SCRLRIT   JMP  SCROLL^RIGHT    ;
9033: 4C EC 91   162   SCRLLFT   JMP  SCROLL^LEFT     ;
9036: 4C 87 91   163   MOVERIGHTJMP  MOVE^RIGHT      ;
9039: 4C 1D 92   164   MOVELEFT  JMP  MOVE^LEFT       ;
903C: 4C 8A 92   165   INVERSE   JMP  SET^INVERSE     ;
903F: 4C AC 92   166   CHHCLBIT  JMP  CHG^COLOR^BIT   ;
9042: 4C CE 92   167   PAGECHNG  JMP  CHANGE^PAGE     ;
9045: 4C F0 92   168   FULLMXD   JMP  TOGGL^FULLMXD   ;
9048: 4C 0E 93   169   PICMERGE  JMP  PICTURE^MERGE   ;
904B: 4C 49 93   170   CLEARSCRNJMP  CLEAR^SCREEN    ;
                 171
                 172
                 173   SCROLL^UP
                 174
                 175
904E: A9 69      176          LDA  #<BUFFER    ;
9050: 85 1C      177          STA  NEWPTR      ;
9052: A9 95      178          LDA  #>BUFFER    ;
9054: 85 1D      179          STA  NEWPTR+1    ;
                 180
9056: A2 00      181          LDX  #TOP^ROW    ;
                 182          >>>  SETPTR.OLDPTR ;
9058: BD 56 93   182          LDA  YLOW,X      ;
905B: 85 1A      182          STA  OLDPTR      ;
905D: 18         182          CLC              ;
905E: BD 16 94   182          LDA  YHIGH,X     ;
9061: 65 19      182          ADC  HRPAGE      ;
9063: 85 1B      182          STA  OLDPTR+1    ;
                 182          <<<              ;
                 183
9065: 20 4E 91   184          JSR  MOVEROW     ;
                 185
                 186
9068: A2 00      187          LDX  #TOP^ROW    ;
906A: 86 1E      188          STX  ROW         ;
```

```
                            >>>   SETPTR.NEWPTR ;
906C:  BD 56 93             LDA   YLOW,X      ;
906F:  85 1C                STA   NEWPTR      ;
9071:  18                   CLC               ;
9072:  BD 16 94             LDA   YHIGH,X     ;
9075:  65 19                ADC   HRPAGE      ;
9077:  85 1D                STA   NEWPTR+1    ;
                            <<<
9079:  E6 1E                INC   ROW         ;
907B:  A6 1E                LDX   ROW         ;
                            >>>   SETPTR.OLDPTR ;
907D:  BD 56 93             LDA   YLOW,X      ;
9080:  85 1A                STA   OLDPTR      ;
9082:  18                   CLC               ;
9083:  BD 16 94             LDA   YHIGH,X     ;
9086:  65 19                ADC   HRPAGE      ;
9088:  85 1B                STA   OLDPTR+1    ;
                            <<<               ;
908A:  20 4E 91    UP:LOOP  JSR   MOVEROW     ;
908D:  A6 1E                LDX   ROW         ;
                            >>>   SETPTR.NEWPTR ;
908F:  BD 56 93             LDA   YLOW,X      ;
9092:  85 1C                STA   NEWPTR      ;
9094:  18                   CLC               ;
9095:  BD 16 94             LDA   YHIGH,X     ;
9098:  65 19                ADC   HRPAGE      ;
909A:  85 1D                STA   NEWPTR+1    ;
                            <<<               ;
909C:  E6 1E                INC   ROW         ;
909E:  A6 1E                LDX   ROW         ;
                            >>>   SETPTR.OLDPTR ;
90A0:  BD 56 93             LDA   YLOW,X      ;
90A3:  85 1A                STA   OLDPTR      ;
90A5:  18                   CLC               ;
90A6:  BD 16 94             LDA   YHIGH,X     ;
90A9:  65 19                ADC   HRPAGE      ;
90AB:  85 1B                STA   OLDPTR+1    ;
                            <<<               ;
90AD:  E0 C0                CPX   #BOT^ROW+1  ;
90AF:  90 D9                BCC   UP:LOOP     ;
90B1:  A9 69                LDA   #<BUFFER    ;
90B3:  85 1A                STA   OLDPTR
90B5:  A9 95                LDA   #>BUFFER
90B7:  85 1B                STA   OLDPTR+1
90B9:  A2 BF                LDX   #BOT^ROW
                            >>>   SETPTR.NEWPTR
90BB:  BD 56 93             LDA   YLOW,X      ;
90BE:  85 1C                STA   NEWPTR      ;
90C0:  18                   CLC               ;
90C1:  BD 16 94             LDA   YHIGH,X     ;
90C4:  65 19                ADC   HRPAGE      ;
90C6:  85 1D                STA   NEWPTR+1    ;
                            <<<               ;
90C8:  20 4E 91             JSR   MOVEROW     ;
90CB:  4C D6 8F             JMP   KEYIN       ;
           SCROLL^DOWN
90CE:  A9 69                LDA   #<BUFFER    ;
90D0:  85 1C                STA   NEWPTR      ;
90D2:  A9 95                LDA   #>BUFFER
90D4:  85 1D                STA   NEWPTR+1
90D6:  A2 BF                LDX   #BOT^ROW    ;
                            >>>   SETPTR.OLDPTR ;
90D8:  BD 56 93             LDA   YLOW,X      ;
90DB:  85 1A                STA   OLDPTR      ;
90DD:  18                   CLC               ;
90DE:  BD 16 94             LDA   YHIGH,X     ;
90E1:  65 19                ADC   HRPAGE      ;
90E3:  85 1B                STA   OLDPTR+1    ;
                            <<<               ;
90E5:  20 4E 91             JSR   MOVEROW     ;
90E8:  A2 BF                LDX   #BOT^ROW    ;
90EA:  86 1E                STX   ROW         ;
                            >>>   SETPTR.NEWPTR
90EC:  BD 56 93             LDA   YLOW,X      ;
90EF:  85 1C                STA   NEWPTR      ;
90F1:  18                   CLC               ;
90F2:  BD 16 94             LDA   YHIGH,X     ;
90F5:  65 19                ADC   HRPAGE      ;
90F7:  85 1D                STA   NEWPTR+1    ;
                            <<<               ;
90F9:  C6 1E                DEC   ROW         ;
90FB:  A6 1E                LDX   ROW         ;
                            >>>   SETPTR.OLDPTR
90FD:  BD 56 93             LDA   YLOW,X      ;
9100:  85 1A                STA   OLDPTR      ;
9102:  18                   CLC               ;
9103:  BD 16 94             LDA   YHIGH,X     ;
9106:  65 19                ADC   HRPAGE      ;
9108:  85 1B                STA   OLDPTR+1    ;
                            <<<               ;
910A:  20 4E 91   DWN:LOOP  JSR   MOVEROW     ;
910D:  A6 1E                LDX   ROW         ;
                            >>>   SETPTR.NEWPTR ;
910F:  BD 56 93             LDA   YLOW,X      ;
9112:  85 1C                STA   NEWPTR      ;
9114:  18                   CLC               ;
9115:  BD 16 94             LDA   YHIGH,X     ;
9118:  65 19                ADC   HRPAGE      ;
911A:  85 1D                STA   NEWPTR+1    ;
                            <<<               ;
911C:  C6 1E                DEC   ROW         ;
911E:  A6 1E                LDX   ROW         ;
                            >>>   SETPTR.OLDPTR
9120:  BD 56 93             LDA   YLOW,X      ;
9123:  85 1A                STA   OLDPTR      ;
9125:  18                   CLC               ;
9126:  BD 16 94             LDA   YHIGH,X     ;
9129:  65 19                ADC   HRPAGE      ;
912B:  85 1B                STA   OLDPTR+1    ;
                            <<<               ;
912D:  E0 FF                CPX   #TOP^ROW-1  ;
912F:  90 D9                BCC   DWN:LOOP    ;
9131:  A9 69                LDA   #<BUFFER    ;
9133:  85 1A                STA   OLDPTR
9135:  A9 95                LDA   #>BUFFER
9137:  85 1B                STA   OLDPTR+1
9139:  A2 00                LDX   #TOP^ROW    ;
                            >>>   SETPTR.NEWPTR
913B:  BD 56 93             LDA   YLOW,X      ;
913E:  85 1C                STA   NEWPTR      ;
9140:  18                   CLC               ;
9141:  BD 16 94             LDA   YHIGH,X     ;
9144:  65 19                ADC   HRPAGE      ;
9146:  85 1D                STA   NEWPTR+1    ;
                            <<<               ;
9148:  20 4E 91             JSR   MOVEROW     ;
```

```
914B: 4C D6 8F                JMP   KEYIN        ;

                MOVEROW
914E: A0 27                   LDY   #NUMCOL      ;
9150: B1 1A     MOVEIT        LDA   (OLDPTR),Y   ;
9152: 91 1C                   STA   (NEWPTR),Y   ;
9154: 88                      DEY                ;
9155: 10 F9                   BPL   MOVEIT       ;
9157: 60                      RTS                ;

                SCROLL^RIGHT

9158: A2 00                   LDX   #TOP^ROW     ;
                RIT:MOVE      >>>   SETPTR.OLDPTR ;
915A: BD 56 93                LDA   YLOW,X       ;
915D: 85 1A                   STA   OLDPTR       ;
915F: 18                      CLC                ;
9160: BD 16 94                LDA   YHIGH,X      ;
9163: 65 19                   ADC   HRPAGE       ;
9165: 85 1B                   STA   OLDPTR+1     ;
                              <<<                ;

9167: A0 27                   LDY   #NUMCOL      ;
9169: B1 1A                   LDA   (OLDPTR),Y   ;
916B: 8D 69 95                STA   BUFFER       ;
916E: 88                      DEY                ;

916F: B1 1A     RIT:LOOP      LDA   (OLDPTR),Y   ;
9171: C8                      INY                ;
9172: 91 1A                   STA   (OLDPTR),Y   ;
9174: 88                      DEY                ;
9175: 88                      DEY                ;
9176: 10 F7                   BPL   RIT:LOOP     ;

9178: A0 00                   LDY   #0           ;
917A: AD 69 95                LDA   BUFFER       ;
917D: 91 1A                   STA   (OLDPTR),Y   ;

917F: E8                      INX                ;
9180: E0 C0                   CPX   #BOT^ROW+1   ;
9182: 90 D6                   BCC   RIT:MOVE     ;
9184: 4C D6 8F                JMP   KEYIN        ;

                MOVE^RIGHT

9187: A2 00                   LDX   #TOP^ROW     ;
                MBR:LOOP      >>>   SETPTR.NEWPTR ;
9189: BD 56 93                LDA   YLOW,X       ;
918C: 85 1C                   STA   NEWPTR       ;
918E: 18                      CLC                ;
918F: BD 16 94                LDA   YHIGH,X      ;
9192: 65 19                   ADC   HRPAGE       ;
9194: 85 1D                   STA   NEWPTR+1     ;
                              <<<                ;

9196: A0 28                   LDY   #NUMCOL+1    ;
9198: A9 00                   LDA   #0           ;
919A: 85 02                   STA   NEWBYTE      ;
919C: 99 69 95  RCLOOP        STA   BUFFER,Y     ;
919F: 88                      DEY                ;
91A0: 10 FA                   BPL   RCLOOP       ;

91A2: A0 00                   LDY   #0           ;
91A4: B1 1C     MBR:SHFT      LDA   (NEWPTR),Y   ;
91A6: 48                      PHA                ;
91A7: 29 80                   AND   #%10000000   ;
91A9: 85 01                   STA   COLORBIT     ;
91AB: 68                      PLA                ;
91AC: 0A                      ASL                ;
91AD: 0A                      ASL                ;
91AE: 26 02                   ROL   NEWBYTE      ;
91B0: 4A                      LSR                ;
91B1: F0 0A                   BEQ   MBR:2        ;
91B3: 19 69 95                ORA   BUFFER,Y     ;
91B6: 29 7F                   AND   #%01111111   ;
91B8: 05 01                   ORA   COLORBIT     ;
91BA: 99 69 95                STA   BUFFER,Y     ;

91BD: C8        MBR:2         INY                ;
91BE: A5 02                   LDA   NEWBYTE      ;
91C0: F0 05                   BEQ   MBR:3        ;
91C2: 05 01                   ORA   COLORBIT     ;
91C4: 99 69 95                STA   BUFFER,Y     ;

91C7: A9 00     MBR:3         LDA   #0           ;
91C9: 85 02                   STA   NEWBYTE      ;

91CB: C0 28                   CPY   #NUMCOL+1    ;
91CD: 90 D5                   BCC   MBR:SHFT     ;

91CF: A0 27                   LDY   #NUMCOL      ;
91D1: B9 69 95  MBLOOP        LDA   BUFFER,Y     ;
91D4: 91 1C                   STA   (NEWPTR),Y   ;
91D6: 88                      DEY                ;
91D7: 10 F8                   BPL   MBLOOP       ;

91D9: A0 28                   LDY   #NUMCOL+1    ;
91DB: B9 69 95                LDA   BUFFER,Y     ;
91DE: A0 00                   LDY   #0           ;
91E0: 11 1C                   ORA   (NEWPTR),Y   ;
91E2: 91 1C                   STA   (NEWPTR),Y   ;

91E4: E8                      INX                ;
91E5: E0 C0                   CPX   #BOT^ROW+1   ;
91E7: 90 A0                   BCC   MBR:LOOP     ;
91E9: 4C D6 8F                JMP   KEYIN        ;

                SCROLL^LEFT

91EC: A2 00                   LDX   #TOP^ROW     ;
                LFT:MOVE      >>>   SETPTR.OLDPTR ;
91EE: BD 56 93                LDA   YLOW,X       ;
91F1: 85 1A                   STA   OLDPTR       ;
91F3: 18                      CLC                ;
91F4: BD 16 94                LDA   YHIGH,X      ;
91F7: 65 19                   ADC   HRPAGE       ;
91F9: 85 1B                   STA   OLDPTR+1     ;
                              <<<                ;

91FB: A0 00                   LDY   #0           ;
91FD: B1 1A                   LDA   (OLDPTR),Y   ;
91FF: 8D 69 95                STA   BUFFER       ;
9202: C8                      INY                ;

9203: B1 1A     LFT:LOOP      LDA   (OLDPTR),Y   ;
9205: 88                      DEY                ;
9206: 91 1A                   STA   (OLDPTR),Y   ;
9208: C8                      INY                ;
9209: C8                      INY                ;
920A: C0 28                   CPY   #NUMCOL+1    ;
920C: 90 F5                   BCC   LFT:LOOP     ;

920E: A0 27                   LDY   #NUMCOL      ;
9210: AD 69 95                LDA   BUFFER       ;
9213: 91 1A                   STA   (OLDPTR),Y   ;

9215: E8                      INX                ;
9216: E0 C0                   CPX   #BOT^ROW+1   ;
9218: 90 D4                   BCC   LFT:MOVE     ;
921A: 4C D6 8F                JMP   KEYIN        ;
```

```
                        MOVE^LEFT
921D: A2 00                       LDX   #TOP^ROW    ;
                        MBL:LOOP  >>>   SETPTR.NEWPTR ;
921F: BD 56 93                    LDA   YLOW,X      ;
9222: 85 1C                       STA   NEWPTR      ;
9224: 18                          CLC               ;
9225: BD 16 94                    LDA   YHIGH,X     ;
9228: 65 19                       ADC   HRPAGE      ;
922A: 85 1D                       STA   NEWPTR+1    ;
                                  <<<               ;
922C: A0 28                       LDY   #NUMCOL+1   ;
922E: A9 00                       LDA   #0          ;
9230: 85 02                       STA   NEWBYTE     ;
9232: 99 69 95          LCLOOP    STA   BUFFER,Y    ;
9235: 88                          DEY               ;
9236: 10 FA                       BPL   LCLOOP      ;
9238: A0 27                       LDY   #NUMCOL     ;
923A: B1 1C             MBL:SHFT  LDA   (NEWPTR),Y  ;
923C: 48                          PHA               ;
923D: 29 80                       AND   #%10000000  ;
923F: 85 01                       STA   COLORBIT    ;
9241: 68                          PLA               ;
9242: 29 7F                       AND   #%01111111  ;
9244: 4A                          LSR               ;
9245: 08                          PHP               ;
9246: 66 02                       ROR   NEWBYTE     ;
9248: 46 02                       LSR   NEWBYTE     ;
924A: 28                          PLP               ;
924B: F0 0A                       BEQ   MBL:2       ;
924D: 19 69 95                    ORA   BUFFER,Y    ;
9250: 29 7F                       AND   #%01111111  ;
9252: 05 01                       ORA   COLORBIT    ;
9254: 99 69 95                    STA   BUFFER,Y    ;
9257: 88                MBL:2     DEY               ;
9258: 10 02                       BPL   MBL:4       ;
925A: A0 28                       LDY   #NUMCOL+1   ;
925C: A5 02             MBL:4     LDA   NEWBYTE     ;
925E: F0 05                       BEQ   MBL:3       ;
9260: 05 01                       ORA   COLORBIT    ;
9262: 99 69 95                    STA   BUFFER,Y    ;
9265: A9 00             MBL:3     LDA   #0          ;
9267: 85 02                       STA   NEWBYTE     ;
9269: C0 28                       CPY   #NUMCOL+1   ;
926B: D0 CD                       BNE   MBL:SHFT    ;
926D: A0 27                       LDY   #NUMCOL     ;
926F: B9 69 95          LBLOOP    LDA   BUFFER,Y    ;
9272: 91 1C                       STA   (NEWPTR),Y  ;
9274: 88                          DEY               ;
9275: 10 F8                       BPL   LBLOOP      ;
9277: A0 28                       LDY   #NUMCOL+1   ;
9279: B9 69 95                    LDA   BUFFER,Y    ;
927C: A0 27                       LDY   #NUMCOL     ;
927E: 11 1C                       ORA   (NEWPTR),Y  ;
9280: 91 1C                       STA   (NEWPTR),Y  ;
9282: E8                          INX               ;
9283: E0 C0                       CPX   #BOT^ROW+1  ;
9285: 90 98                       BCC   MBL:LOOP    ;
9287: 4C D6 8F                    JMP   KEYIN

                        SET^INVERSE
928A: A2 00                       LDX   #TOP^ROW    ;
                        INVLOOP   >>>   SETPTR.NEWPTR
928C: BD 56 93                    LDA   YLOW,X      ;
928F: 85 1C                       STA   NEWPTR      ;
9291: 18                          CLC               ;
9292: BD 16 94                    LDA   YHIGH,X     ;
9295: 65 19                       ADC   HRPAGE      ;
9297: 85 1D                       STA   NEWPTR+1    ;
                                  <<<               ;
9299: A0 27                       LDY   #NUMCOL     ;
929B: B1 1C             ROWLOOP   LDA   (NEWPTR),Y  ;
929D: 49 FF                       EOR   #%11111111  ;
929F: 91 1C                       STA   (NEWPTR),Y  ;
92A1: 88                          DEY               ;
92A2: 10 F7                       BPL   ROWLOOP     ;
92A4: E8                          INX               ;
92A5: E0 C0                       CPX   #BOT^ROW+1  ;
92A7: 90 E3                       BCC   INVLOOP     ;
92A9: 4C D6 8F                    JMP   KEYIN       ;

                        CHG^COLOR^BIT
92AC: A2 00                       LDX   #TOP^ROW    ;
                        CBLOOP    >>>   SETPTR.NEWPTR
92AE: BD 56 93                    LDA   YLOW,X      ;
92B1: 85 1C                       STA   NEWPTR      ;
92B3: 18                          CLC               ;
92B4: BD 16 94                    LDA   YHIGH,X     ;
92B7: 65 19                       ADC   HRPAGE      ;
92B9: 85 1D                       STA   NEWPTR+1    ;
                                  <<<               ;
92BB: A0 27                       LDY   #NUMCOL     ;
92BD: B1 1C             RLOOP     LDA   (NEWPTR),Y  ;
92BF: 49 80                       EOR   #%10000000  ;
92C1: 91 1C                       STA   (NEWPTR),Y  ;
92C3: 88                          DEY               ;
92C4: 10 F7                       BPL   RLOOP       ;
92C6: E8                          INX               ;
92C7: E0 C0                       CPX   #BOT^ROW+1  ;
92C9: 90 E3                       BCC   CBLOOP      ;
92CB: 4C D6 8F                    JMP   KEYIN       ;

                        CHANGE^PAGE
92CE: A5 19                       LDA   HRPAGE      ;
92D0: D0 0D                       BNE   MAKE^P1     ;
92D2: 2C 52 C0                    BIT   FULLSCRN    ;
92D5: 2C 55 C0                    BIT   FLIP2       ;
92D8: A9 20                       LDA   #$20        ;
92DA: 85 19                       STA   HRPAGE      ;
92DC: 4C D6 8F                    JMP   KEYIN       ;
92DF: A5 00             MAKE^P1   LDA   FMFLAG      ;
92E1: D0 03                       BNE   HRP:1       ;
92E3: 2C 53 C0                    BIT   MXEDSCRN    ;
92E6: A9 00             HRP:1     LDA   #0          ;
92E8: 85 19                       STA   HRPAGE      ;
92EA: 2C 54 C0                    BIT   FLIP1       ;
92ED: 4C D6 8F                    JMP   KEYIN

                        TOGGL^FULLMXD
92F0: A5 19                       LDA   HRPAGE      ;
92F2: F0 06                       BEQ   TOGGLE      ;
92F4: 20 3A FF                    JSR   BELL        ;
92F7: 4C D6 8F                    JMP   KEYIN
```

```
                    514
92FA: A5 00         515  TOGGLE     LDA  FMFLAG     ;
92FC: 49 01         516             EOR  #1         ;
92FE: 85 00         517             STA  FMFLAG     ;
9300: D0 06         518             BNE  MAKEFULL   ;
                    519
9302: 2C 53 C0      520             BIT  MXEDSCRN   ;
9305: 4C D6 8F      521             JMP  KEYIN
                    522
9308: 2C 52 C0      523  MAKEFULL   BIT  FULLSCRN   ;
930B: 4C D6 8F      524             JMP  KEYIN
                    525
                    526
                    527  PICTURE^MERGE
                    528
930E: A2 00         529             LDX  #TOP^ROW   ;
                    530  MLOOP      >>>  SETPTR.NEWPTR
9310: BD 56 93      530             LDA  YLOW,X     ;
9313: 85 1C         530             STA  NEWPTR     ;
9315: 18            530             CLC             ;
9316: BD 16 94      530             LDA  YHIGH,X    ;
9319: 65 19         530             ADC  HRPAGE     ;
931B: 85 1D         530             STA  NEWPTR+1   ;
                    530             <<<             ;
931D: A5 19         531             LDA  HRPAGE     ;
931F: 49 20         532             EOR  #$20       ;
9321: 85 19         533             STA  HRPAGE     ;
                    534             >>>  SETPTR.OLDPTR ;
9323: BD 56 93      534             LDA  YLOW,X     ;
9326: 85 1A         534             STA  OLDPTR     ;
9328: 18            534             CLC             ;
9329: BD 16 94      534             LDA  YHIGH,X    ;
932C: 65 19         534             ADC  HRPAGE     ;
932E: 85 1B         534             STA  OLDPTR+1   ;
                    534             <<<             ;
9330: A0 27         535             LDY  #NUMCOL    ;
9332: B1 1A         536  RMLOOP     LDA  (OLDPTR),Y ;
9334: 51 1C         537             EOR  (NEWPTR),Y ;
9336: 91 1C         538             STA  (NEWPTR),Y ;
9338: 88            539             DEY             ;
9339: 10 F7         540             BPL  RMLOOP     ;
                    541
933B: A5 19         542             LDA  HRPAGE     ;
933D: 49 20         543             EOR  #$20       ;
933F: 85 19         544             STA  HRPAGE     ;
9341: E8            545             INX             ;
9342: E0 C0         546             CPX  #BOT^ROW+1 ;
9344: 90 CA         547             BCC  MLOOP      ;
9346: 4C D6 8F      548             JMP  KEYIN      ;
                    549
                    550
                    551  CLEAR^SCREEN
                    552
9349: A5 19         553             LDA  HRPAGE     ;
934B: 18            554             CLC             ;
934C: 69 20         555             ADC  #$20       ;
934E: 85 E6         556             STA  HPAGE      ;
9350: 20 F2 F3      557             JSR  HCLR       ;
9353: 4C D6 8F      558             JMP  KEYIN      ;
                    559
                    560
9356: 00 00 00
9359: 00 00 00
935C: 00 00         561  YLOW       HEX  0000000000000000
935E: 80 80 80
9361: 80 80 80
9364: 80 80         562             HEX  8080808080808080
9366: 00 00 00
9369: 00 00 00
936C: 00 00         563             HEX  0000000000000000
936E: 80 80 80
9371: 80 80 80
9374: 80 80         564             HEX  8080808080808080
9376: 00 00 00
9379: 00 00 00
937C: 00 00         565             HEX  0000000000000000
937E: 80 80 80
9381: 80 80 80
9384: 80 80         566             HEX  8080808080808080
9386: 00 00 00
9389: 00 00 00
938C: 00 00         567             HEX  0000000000000000
938E: 80 80 80
9391: 80 80 80
9394: 80 80         568             HEX  8080808080808080
9396: 28 28 28
9399: 28 28 28
939C: 28 28         569             HEX  2828282828282828
939E: A8 A8 A8
93A1: A8 A8 A8
93A4: A8 A8         570             HEX  A8A8A8A8A8A8A8A8
93A6: 28 28 28
93A9: 28 28 28
93AC: 28 28         571             HEX  2828282828282828
93AE: A8 A8 A8
93B1: A8 A8 A8
93B4: A8 A8         572             HEX  A8A8A8A8A8A8A8A8
93B6: 28 28 28
93B9: 28 28 28
93BC: 28 28         573             HEX  2828282828282828
93BE: A8 A8 A8
93C1: A8 A8 A8
93C4: A8 A8         574             HEX  A8A8A8A8A8A8A8A8
93C6: 28 28 28
93C9: 28 28 28
93CC: 28 28         575             HEX  2828282828282828
93CE: A8 A8 A8
93D1: A8 A8 A8
93D4: A8 A8         576             HEX  A8A8A8A8A8A8A8A8
93D6: 50 50 50
93D9: 50 50 50
93DC: 50 50         577             HEX  5050505050505050
93DE: D0 D0 D0
93E1: D0 D0 D0
93E4: D0 D0         578             HEX  D0D0D0D0D0D0D0D0
93E6: 50 50 50
93E9: 50 50 50
93EC: 50 50         579             HEX  5050505050505050
93EE: D0 D0 D0
93F1: D0 D0 D0
93F4: D0 D0         580             HEX  D0D0D0D0D0D0D0D0
93F6: 50 50 50
93F9: 50 50 50
93FC: 50 50         581             HEX  5050505050505050
93FE: D0 D0 D0
9401: D0 D0 D0
9404: D0 D0         582             HEX  D0D0D0D0D0D0D0D0
9406: 50 50 50
9409: 50 50 50
940C: 50 50         583             HEX  5050505050505050
940E: D0 D0 D0
9411: D0 D0 D0
9414: D0 D0         584             HEX  D0D0D0D0D0D0D0D0
9416: 20 24 28
9419: 2C 30 34
941C: 38 3C         585  YHIGH      HEX  2024282C3034383C
941E: 20 24 28
9421: 2C 30 34
9424: 38 3C         586             HEX  2024282C3034383C
9426: 21 25 29
9429: 2D 31 35
942C: 39 3D         587             HEX  2125292D3135393D
942E: 21 25 29
9431: 2D 31 35
9434: 39 3D         588             HEX  2125292D3135393D
9436: 22 26 2A
9439: 2E 32 36
943C: 3A 3E         589             HEX  22262A2E32363A3E
```

```
943E: 22 26 2A
9441: 2E 32 36
9444: 3A 3E      590            HEX   22262A2E32363A3E
9446: 23 27 2B
9449: 2F 33 37
944C: 3B 3F      591            HEX   23272B2F33373B3F
944E: 23 27 2B
9451: 2F 33 37
9454: 3B 3F      592            HEX   23272B2F33373B3F
9456: 20 24 28
9459: 2C 30 34
945C: 38 3C      593            HEX   2024282C3034383C
945E: 20 24 28
9461: 2C 30 34
9464: 38 3C      594            HEX   2024282C3034383C
9466: 21 25 29
9469: 2D 31 35
946C: 39 3D      595            HEX   2125292D3135393D
946E: 21 25 29
9471: 2D 31 35
9474: 39 3D      596            HEX   2125292D3135393D
9476: 22 26 2A
9479: 2E 32 36
947C: 3A 3E      597            HEX   22262A2E32363A3E
947E: 22 26 2A
9481: 2E 32 36
9484: 3A 3E      598            HEX   22262A2E32363A3E
9486: 23 27 2B
9489: 2F 33 37
948C: 3B 3F      599            HEX   23272B2F33373B3F
948E: 23 27 2B
9491: 2F 33 37
9494: 3B 3F      600            HEX   23272B2F33373B3F
9496: 20 24 28
9499: 2C 30 34
949C: 38 3C      601            HEX   2024282C3034383C
949E: 20 24 28
94A1: 2C 30 34
94A4: 38 3C      602            HEX   2024282C3034383C
94A6: 21 25 29
94A9: 2D 31 35
94AC: 39 3D      603            HEX   2125292D3135393D
94AE: 21 25 29
94B1: 2D 31 35
94B4: 39 3D      604            HEX   2125292D3135393D
94B6: 22 26 2A
94B9: 2E 32 36
94BC: 3A 3E      605            HEX   22262A2E32363A3E
94BE: 22 26 2A
94C1: 2E 32 36
94C4: 3A 3E      606            HEX   22262A2E32363A3E
94C6: 23 27 2B
94C9: 2F 33 37
94CC: 3B 3F      607            HEX   23272B2F33373B3F
94CE: 23 27 2B
94D1: 2F 33 37
94D4: 3B 3F      608            HEX   23272B2F33373B3F
                 609
                 610
94D6: D3 C3 D2
94D9: CF CC CC
94DC: A0 C3 CF
94DF: CD CD C1
94E2: CE C4 D3
94E5: BA A0      611 SCLCMNDS ASC   "SCROLL COMMANDS: "
94E7: 3C 2D      612            INV   "<-"
94E9: AC A0      613            ASC   ", "
94EB: 2D 3E      614            INV   "->"
94ED: AC A0      615            ASC   ", "
94EF: 3C         616            INV   "<"
94F0: AC A0      617            ASC   ", "
94F2: 3E         618            INV   ">"
94F3: AC A0      619            ASC   ", "
94F5: 01         620            INV   "A"
94F6: AC A0      621            ASC   ", "
94F8: 1A         622            INV   "Z"
94F9: 00         623            BRK
94FA: 09         624 INV:I      INV   "I"
94FB: BA C9 CE
94FE: D6 C5 D2
9501: D3 C5 A0
9504: C3 CF CC
9507: CF D2 D3   625            ASC   ":INVERSE COLORS"
950A: 00         626            BRK
950B: 03         627 CHGBIT:C   INV   "C"
950C: BA C3 C8
950F: C1 CE C7
9512: C5 A0 C3
9515: CF CC CF
9518: D2 A0 C2
951B: C9 D4      628            ASC   ":CHANGE COLOR BIT"
951D: 00         629            BRK
951E: 06         630 FLMX:F     INV   "F"
951F: BA C6 D5
9522: CC CC AF
9525: CD C9 D8
9528: C5 C4 A0
952B: D4 CF C7
952E: C7 CC C5   631            ASC   ":FULL/MIXED TOGGLE"
9531: 00         632            BRK
9532: 0D         633 MERGE:M    INV   "M"
9533: BA CD C5
9536: D2 C7 C5
9539: A0 D3 C3
953C: D2 C5 C5
953F: CE D3      634            ASC   ":MERGE SCREENS"
9541: 00         635            BRK
9542: 03 14 12
9545: 0C 2D 20   636 CLEAR:@    INV   "CTRL- "
9548: BA C3 CC
954B: C5 C1 D2
954E: A0 D3 C3
9551: D2 C5 C5
9554: CE         637            ASC   ":CLEAR SCREEN"
9555: 00         638            BRK
9556: 10         639 PAGE:P     INV   "P"
9557: BA C6 CC
955A: C9 D0 A0
955D: D0 C1 C7
9560: C5         640            ASC   ":FLIP PAGE"
9561: 00         641            BRK
9562: 11         642 QUIT:Q     INV   "Q"
9563: BA D1 D5
9566: C9 D4      643            ASC   ":QUIT"
9568: 00         644            BRK
                 645
                 646 BUFFER     DS    1
                 647

--End assembly--

1642 bytes

Errors: 0
```

程式的設計與FORTRAN所帶來的好處 雙精度模式是其中一項優異功能

大家都知道，保證電腦完成某一預定任務（例如解題或控制某一過程）時，人爲編排的指令序列或語句序列就叫做程序（或程式），把預定任務用程序實現的全部過程叫做程序設計。在我們的環境之中，程序設計其實都是一種語言翻譯程序，一般分爲滙編程序和編譯程序。前都是一對一地把滙編語言（或稱爲組合語言）譯成機械語言程式，而編譯程序是一對多地把高級語言程序譯成機器語言程序，於是我們會問：究竟選擇何者爲佳？如果兩者的軟件支持都有保證的話。

兩個問題

考慮的角度有兩個，第一是程式對硬件的依賴程度。第二是工作量的問題。現在先就第一個問題來說，使用滙編程序，設計者必須對機器本身的體系結構瞭如指掌，編寫程式時必須以滙編語言分配全部資源，正確安排每個寄存器的操作。而使用編譯程序，可以用通用的語言編寫程序，而幾乎不需考慮機器本身的結構。

對第二個問題（工作量）來說，我們都明白用于電腦的用戶應用程序，一般約包含200至4000條指令句，程序規模越大，用滙編語言編程序的工作量要比用高級語言多30～5000%。事實證明，產生一個無錯誤程序的機率是會隨著程序規模（長度）和複雜性的提高而指數式地下降。可見滙編語言用來產生大程序是不適合的，特別對於缺乏經驗的程序員來說更是如此。一個典型的電腦程式編寫指數最能說明這個問題，這個指數就是；對于調試好的目的碼（OBJECT CODE），滙編語言是

5條指令／程序員日；

而高級語言是

100條指令／程序員日。

由於這個原故我們馬上會考慮到程序效率問題，程序效率一般是指完成同一解題過程所需目標程序的規模或佔用的存儲空間（長度），另一方面也可能是指程序的執行速度。用高級語言編寫

列表(1)

```
B>type pi.for
C
C    PI.FOR - THIS IS THE SOURCE PROGRAM
C
C    Program & Documentation by Ng Shiu Wah
C
C   THIS PROGRAM CALCULATES AN ESTIMATE
C   OF PI BY CALCULATING THE CIRCUMFERENCE
C   OF A REGULAR POLYGON, AND DIVIDING THIS
C   BY THE RADIUS OF THE POLYGON.
C
        PROGRAM PI

        DOUBLE PRECISION MINPI, MAXPI, SLNGTH, FACT
        DOUBLE PRECISION K, SIDES, SUM, SSQ, TERM, TEMP

        WRITE(3,100)

        SIDES = 4.0
        SUM   = 2.0

        DO 40 N = 3,20

10      SIDES = 2.0 * SIDES
        SSQ   = SUM
        SUM   = 0.0
        TERM  = SSQ / 4.0
        K     = 1

20      TEMP = TERM + SUM

        IF (TEMP .LE. SUM) GO TO 30

        SUM  = TEMP
        FACT = (2 * K - 1)  / (K + 1)
        TERM = FACT * SSQ * TERM / 8.0

        K    = K + 1
        GO TO 20

30      SLNGTH = DSQRT(SUM)

        TEMP   = SIDES * SLNGTH
        MINPI  = TEMP / 2.0
        MAXPI  = TEMP / (2.0 - SLNGTH)

40      WRITE(3,200) SIDES, MINPI, MAXPI

        STOP * PI *

100     FORMAT('1BOUNDS ON PI - BINOMIAL THEOREM VERSION'//
     +  '  SIDES        LOWER BOUND      UPPER BOUND'//)

200     FORMAT(1X,F8.0,1X,F16.14,1X,F16.14)

        END

B>
```

程式的設計與 FORTRAN 所帶來的好處

的程序是在一對多的基礎上譯成目標程序的，一條原程序指令（SOURCE FILE COMMAND）可能譯成幾條，甚至幾十條的目標程序。由于高級語言的語法結構要照顧通用性，因此機器本身的設計特點就沒有充份利用，因而經編譯或解釋而得到的機器執行指令總是要比用滙編語言直接得到的指令數多得多。一般在最佳情況下也要多10%—20%的指令數。否則會超出30—50%。程序指令數增多，自然會在同一硬件條件下降低程序執行的速度。因此，如果考慮微計算機儲存容量較小而必須提高程序效率的時候，還是採用滙編語計來編寫較好。不過由於近年來電腦儲存的容量大增而價格下降，上面的矛盾也就逐步消除，因此若果硬件性能高的話，採用高級語言是有利的。

一個熟練的程序員通常是僅僅熟練地掌握一種語言來編寫程序，熟練各種語言的程序員通常都很難找到，所以對於特定的用戶系統，究竟使用那種語言編程序，一方面要取决於軟件支持，另一方面你還得要看程序員的能力，不過話說回來，若果我們要求電腦處理精確計算的問題時，採用FORTRAN 來編寫會較爲恰當。

FORTRAN 的選擇

FORTRAN是FORMULA TRANSLATION 的縮寫，直譯過來就是"公式翻譯程序語言"，它是最早產生編譯程序的高級語言之一。在五十年代後期在IBM 704機上試驗的FORTRAN I證實了編譯程序完全可以產生有效的目標程序（OBJECT PROGRAM），而且證明它事實上是建立大型科學計算和工程計算程序的有力工具。隨後不久發展了FORTRAN II，其中消除了一些無用的語言屬性，並且加進了重要的子程序概念。沒有人知道甚麽時候搞了FORTRAN III，後來就出現了FORTRAN IV，並且成爲國際標準。在1978年更修定改成爲FORTRAN 77 新標準。在一般工程或精確計算上，都採用FORTRAN 完成，在大學裏一般工科學生幾乎在學校只學到FORTRAN 語言，以往許多程序員也唯一地只熟悉這種語言（包括筆者在內）。我們今天常用的BASIC就是從FORTRAN 脫胎出

```
B>
B>type pi.prn
FORTRAN-80 Ver. 3.43 Copyright 1978-1981 (C) By Microsoft -- Bytes: 25131
Created: 27-Jun-81
1       C
2       C      PI.FOR - THIS IS THE SOURCE PROGRAM
3       C
4       C      Program & Documentation by Ng Shiu Wah
5       C
6       C      THIS PROGRAM CALCULATES AN ESTIMATE
7       C      OF PI BY CALCULATING THE CIRCUMFERENCE
8       C      OF A REGULAR POLYGON, AND DIVIDING THIS
9       C      BY THE RADIUS OF THE POLYGON.
10      C
11                 PROGRAM PI
12
13                 DOUBLE PRECISION MINPI, MAXPI, SLNGTH, FACT
*****   0000'      LD        BC,$$L
*****   0003'      JP        $INIT
14                 DOUBLE PRECISION K, SIDES, SUM, SSQ, TERM, TEMP
15
16                 WRITE(3,100)
*****   0006'      LD        DE,100L
*****   0009'      LD        HL,[03 00 00 00]
*****   000C'      CALL      $W2
17
18                 SIDES = 4.0
*****   000F'      CALL      $ND
19                 SUM   = 2.0
20
21                 DO 40 N = 3,20
*****   0012'      LD        HL,[00 00 00 83]
*****   0015'      CALL      $L1
*****   0018'      CALL      $CK
*****   001B'      LD        HL,SIDES
*****   001E'      CALL      $T3
*****   0021'      LD        HL,[00 00 00 82]
*****   0024'      CALL      $L1
*****   0027'      CALL      $CK
*****   002A'      LD        HL,SUM
*****   002D'      CALL      $T3
22
23      10         SIDES = 2.0 * SIDES
*****   0030'      LD        HL,0003
*****   0033'      LD        (N),HL
24                 SSQ   = SUM
25                 SUM   = 0.0
26                 TERM  = SSQ / 4.0
27                 K     = 1
28
29      20         TEMP = TERM + SUM
*****   0036'      LD        HL,[00 00 00 82]
*****   0039'      CALL      $L1
*****   003C'      CALL      $CK
*****   003F'      LD        HL,SIDES
*****   0042'      CALL      $MU
*****   0045'      LD        HL,SIDES
*****   0048'      CALL      $T3
*****   004B'      LD        HL,SUM
*****   004E'      CALL      $L3
*****   0051'      LD        HL,SSQ
*****   0054'      CALL      $T3
*****   0057'      LD        HL,[00 00 00 00]
*****   005A'      CALL      $L1
*****   005D'      CALL      $CK
*****   0060'      LD        HL,SUM
*****   0063'      CALL      $T3
*****   0066'      LD        HL,SSQ
*****   0069'      CALL      $L3
*****   006C'      LD        HL,[00 00 00 83]
*****   006F'      CALL      $DR
*****   0072'      LD        HL,TERM
*****   0075'      CALL      $T3
*****   0078'      LD        HL,0001
*****   007B'      CALL      $CC
*****   007E'      LD        HL,K
*****   0081'      CALL      $T3
```

列表(2)

程式的設計與 FORTRAN 所帶來的好處

來的，因此用這種語言寫的程序，形式上有些像BASIC，但FORTRAN 的語法和語句含義都要複雜得多，而且它的語句不需要標行號，交談性能極高（HIGH INTERACTIVE）。

FORTRAN 幾乎總是配備編譯程序，因此通常只在大型電腦上才可享有，最主要的原因是它的語言功能複雜，編譯程序的規模約 BASIC 大一倍左右，換句話說要佔用頗大的記憶體來運作，這對昔日的微電腦來說是極為困難的。不過在10年前以8080為CPU 的微電腦上開始已能應用FORTRAN 的一個子集，（INTEL 公司發展），後來又在8080，8085，8086上實現了FORT-RAN 77 的子集，而現時的Z80 CPU 基本上是8080的改良擴展形態，因此亦享受到以上的成果。在目前更多的廠商都是用其它高級語言的子集編寫微型電腦的原程式，再用 FORTRAN Ⅳ 形成交義編譯程序，以便利使用大型電腦中普遍擁有的，用於FORTRAN 的軟件資源。

現時我們使用在我們微電腦系統中

```
                 IF (TEMP .LE. SUM) GO TO 30
*****   0084'   LD      HL,TERM
*****   0087'   CALL    $L3
*****   008A'   LD      HL,SUM
*****   008D'   CALL    $AU
*****   0090'   LD      HL,TEMP
*****   0093'   CALL    $T3
*****   0096'   LD      HL,SUM
*****   0099'   CALL    $SU
*****   009C'   DEC     A
*****   009D'   ADD     81
*****   009F'   SBC     A
*****   00A0'   LD      (T:000002),A

                SUM  = TEMP
*****   00A3'   OR      A
*****   00A4'   JP      NZ,30L
                FACT = (2 * K - 1)  / (K + 1)
                TERM = FACT * SSQ * TERM / 8.0

                K    = K + 1
                GO TO 20
*****   00A7'   LD      HL,TEMP
*****   00AA'   CALL    $L3
*****   00AD'   LD      HL,SUM
*****   00B0'   CALL    $T3
*****   00B3'   LD      HL,K
*****   00B6'   CALL    $L3
*****   00B9'   LD      HL,0001
*****   00BC'   CALL    $AQ
*****   00BF'   LD      HL,T:000003
*****   00C2'   CALL    $T3
*****   00C5'   LD      HL,K
*****   00C8'   CALL    $L3
*****   00CB'   LD      HL,0002
*****   00CE'   CALL    $MQ
*****   00D1'   LD      HL,0001
*****   00D4'   CALL    $SQ
*****   00D7'   LD      HL,T:000003
*****   00DA'   CALL    $DU
*****   00DD'   LD      HL,FACT
*****   00E0'   CALL    $T3
*****   00E3'   LD      HL,TERM
*****   00E6'   CALL    $L3
*****   00E9'   LD      HL,SSQ
*****   00EC'   CALL    $MU
*****   00EF'   LD      HL,FACT
*****   00F2'   CALL    $MU
*****   00F5'   LD      HL,[00 00 00 84]
*****   00F8'   CALL    $DR
*****   00FB'   LD      HL,TERM
*****   00FE'   CALL    $T3
*****   0101'   LD      HL,T:000003
*****   0104'   CALL    $L3
*****   0107'   LD      HL,K
*****   010A'   CALL    $T3

        30      SLNGTH = DSQRT(SUM)
*****   010D'   JP      20L

                TEMP   = SIDES * SLNGTH
                MINPI  = TEMP / 2.0
                MAXPI  = TEMP / (2.0 - SLNGTH)

        40      WRITE(3,200) SIDES, MINPI, MAXPI
*****   0110'   LD      HL,SUM
*****   0113'   CALL    DSQRT
*****   0116'   LD      HL,SLNGTH
*****   0119'   CALL    $T3
*****   011C'   LD      HL,SIDES
*****   011F'   CALL    $L3
*****   0122'   LD      HL,SLNGTH
*****   0125'   CALL    $MU
*****   0128'   LD      HL,TEMP
*****   012B'   CALL    $T3
*****   012E'   LD      HL,[00 00 00 82]
*****   0131'   CALL    $DR
*****   0134'   LD      HL,MINPI
*****   0137'   CALL    $T3
*****   013A'   LD      HL,[00 00 00 82]
*****   013D'   CALL    $L1
*****   0140'   CALL    $CK
*****   0143'   LD      HL,SLNGTH
*****   0146'   CALL    $SU
*****   0149'   LD      HL,T:000003
*****   014C'   CALL    $T3
*****   014F'   LD      HL,TEMP
*****   0152'   CALL    $L3
*****   0155'   LD      HL,T:000003
*****   0158'   CALL    $DU
*****   015B'   LD      HL,MAXPI
*****   015E'   CALL    $T3
*****   0161'   LD      DE,200L
*****   0164'   LD      HL,[03 00 00 00]
*****   0167'   CALL    $W2

                STOP * PI *
*****   016A'   LD      BC,005C"
*****   016D'   LD      DE,SIDES
*****   0170'   LD      HL,[01 00 00 00]
*****   0173'   LD      A,04
*****   0175'   CALL    $I3
*****   0178'   CALL    $ND
*****   017B'   LD      HL,(N)
*****   017E'   INC     HL
*****   017F'   LD      A,14
*****   0181'   SUB     L
*****   0182'   LD      A,00
*****   0184'   SBC     H
*****   0185'   JP      P,0033'
*****   0188'   CALL    $ST

        100     FORMAT('1BOUNDS ON PI - BINOMIAL THEOREM VERSION'//
*****   018B'   2A50492A2020
             +  '   SIDES        LOWER BOUND     UPPER BOUND'//)

        200     FORMAT(1X,F8.0,1X,F16.14,1X,F16.14)

                END
*****   0191'   00 00 00 00
*****   0195'   01 00 00 00
*****   0199'   03 00 00 00
*****   019D'   00 00 00 83
*****   01A1'   00 00 00 82
*****   01A5'   00 00 00 84

Program Unit Length=01A9 (425) Bytes
Data Area Length=00D6 (214) Bytes

Subroutines Referenced:

$I3                   DSQRT                 $INIT
$W2                   $ND                   $L1
$CK                   $T3                   $MU
$L3                   $DR                   $CC
$AU                   $SU                   $AQ
$MQ                   $SQ                   $DU
$ST

Variables:

MINPI    0001"        MAXPI    0009"        SLNGTH   0011"
FACT     0019"        K        0021"        SIDES    0029"
SUM      0031"        SSQ      0039"        TERM     0041"
TEMP     0049"        N        0051"        T:000002        0053"
T:000003        0054"

Labels:

$$L      0006'        100L     0060"        40L      0161'
10L      0036'        20L      0084'        30L      0110'
200L     00B9"
```

程式的設計與FORTRAN所帶來的好處

的FORTRAN 版本是FORTRAN 80，80的意思是以Z80爲CPU 的電腦系統上運行的FORTRAN 版本，在1980年由MICROSOFT 公司推出的CP／M語言軟件，基本上只要具有Z80咭在SLOT 4及16K RAM 咭在SLOT上即可運行，當然其它電腦上只要可以運行CP／M的亦可運行無誤。

FORTRAN—80 具備了強有力的指令群，只要程式設計良好，均可和IBM，CDC等大型電腦相容，換句話說，在FORTRAN—80上運行的程式，稍爲將少許MACHINE—DEPENDENT的指令修改一下，即可在大型電腦上執行，同樣地大型電腦上所構建的FORT—RAN程式，只要稍加配合FORTRAN—80的指令修改，也可以在IBMPC，APPLE，NEC或任何可運行CP／M的機器上執行。

最後筆者以一種FORTRAN 的優異性能DOUBLE PRECISION 雙倍精度模式的引用作爲結束，以後我會爲各位再深入探討FORTRAN。

雙倍精度在運算程式上的好處是容許以16位（包括符號和小數點）作運算，俾能增加數字的精確度，例如—56789.83542155774。而一般語言是能提供SINGLE PRECISION的運算結果，例如—999887。就是，比較上來說，在要求精度不高的環境下還可接受，但在較高的要求下則會一籌莫展了，以下的程式①是計算PI值的示範程式（原程式），列表②是該程式的編譯程式，而結果的目標程式則由於篇幅所限不能刊出。（灌錄於程式磁碟14B上）（檔案名稱爲PI.COM）

執行的結果見列表③。■

列表(3)

```
B>pi
BOUNDS ON PI - BINOMIAL THEOREM VERSION

 SIDES     LOWER BOUND       UPPER BOUND


      8. 3.06146745892072 4.95931523537420
     16. 3.12144515225805 3.87800673496263
     32. 3.13654849054594 3.47739256563251
     64. 3.14033115695475 3.30237081249040
    128. 3.14127725093277 3.22030755454287
    256. 3.14151380114430 3.18054396821973
    512. 3.14157294036709 3.16096827709498
   1024. 3.14158772527716 3.15125564133382
   2048. 3.14159142151120 3.14641796432625
   4096. 3.14159234557012 3.14400376602075
   8192. 3.14159257658487 3.14279782442605
  16384. 3.14159263433856 3.14219514270746
  32768. 3.14159264877699 3.14189387407905
  65536. 3.14159265238659 3.14174325781772
 131072. 3.14159265328899 3.14166795419967
 262144. 3.14159265351459 3.14163030351872
 524288. 3.14159265357099 3.14161147846025
1048576. 3.14159265358509 3.14160206600152 STOP   *PI*
```


(1) CP／M的密碼檔案
(2) FORTRAN所帶來的好處
(3) 令蘋果發言的最佳工具
(4) 郵費計算諮詢系統

收錄在一張磁碟上，編號14B，讀者服務部有售，每張29.90。或利用郵購，郵購請加1元郵費。

中四、中五程度同學適宜參考

英文字義辨識練習

在上期中筆者將達爾文的進化論試用電腦來進行模擬，讓同學們能有一個更具體的認識。編寫這種程式並不容易。今期筆者由於時間所限，就地取材寫了一個結構簡單但非常有用的程式，供測試同學們對英文字義認識的程度。大家可以將之視爲一個遊戲亦無不可。老師們亦可將之收集作爲日後的教材。

當執行本程式時，螢幕會出現一個MENU。同學們可透過印字機或螢幕進行。遊戲會先打出一句的定義，然後有三個近似的生字答案供選擇，玩者要選出正確的生字。

開始時，電腦會詢問要答多少條題目。遊戲進行時會將答對的題目記錄，並於題目發問完後，顯示出所答中的題數，然後給予評語。

同學可以大家之間進行比試，看誰識生字最多。

本遊戲程式共包括兩個FILE，其一是主程式的APPLE—SOFT，第二個是放生字題目的TEXT FILE。

由於這個程式著重於同學對英文生字的認識而且有一個TEXT FILE 裝載生字，若要將程式打入則沒有多大意義。所以筆者不擬將這兩個FILE 列出。同學們可在今期的電腦時代程式磁碟上找到兩個FILE：SPELLING DRILLS及SPELLING WORDS。

SPELLING WORDS 檔案內放入了61個生字，電腦會隨機選擇。日後有機會筆者再增加這個檔案的容量。大家不妨試試能否正確辨識全部的生字。■

MBASIC 譯碼/解碼儀

防範檔案內容遭人窺看

王國良

假如你精心編寫的程序放在學校的磁碟庫中時，你會否担心它會給別人抄去？你用甚麽方法使你的程序或文字檔案不會給人參閱？譯碼（ENCRYPTION）和解碼（DECRYPTION）就是解決以上問題的最佳方法。

在未介紹譯碼的方法和程序之前，先要引出幾過名詞：

密碼鎖（CIPHER）——一些方法使原始檔案譯成密碼，使外人不能參閱。

KEY ——可以直接解作鎖匙，用來開啓密碼鎖。

本文檔案（PLAIN TEXT）——未加上密碼鎖的文字檔案。

密碼文件（CIPHER TEXT）——加上了密碼鎖的文件。

今次使用的譯碼方法，是一九一七年由一名工程司設計出來的。

它的工作方法大致如下：

①把文字轉變成一連串的 BITS。

②產生一串隨機的 BITS，這就是 KEY。

③將原本的文字和 KEY作一個 XOR（EXCLUSIVE—OR），產生出來的就是密碼文件。

讓我們看看電腦怎樣依據上述的步驟來翻譯我們的文件吧！

步驟①電腦是已經替我們做了，因爲在電腦上，所有的通訊都是以 BIT 爲單位的。

步驟②的 KEY，相等於軍營中的暗語或開門的鎖匙，是全部譯碼過程中最重要的部份。

步齒③是將本文檔案 XOR KEY 產生密碼，可見圖一。

我們想把盜竊我們文件的人稱爲敵方。在譯碼時，我們假設敵方已經知道了我們翻譯的方法，而保護密碼文件的就只靠一個 KEY了。如果這個 KEY是一個絕對的隨機數（ABSOLUTE RANDOM），在理論上是無方解破的。可惜，在現實生活上，我們能夠記到的長度有限，而一個短的KEY 是會很容易給專家解破的。

要產生一個長的 KEY最簡單的方法就是把幾個短的KEY 混合一起，它的排列次序在第一循環內是不是相同的，例如：

```
(a)  APPLEAPPLEAPPLEAPPLEAPPLEAPPLEAPPLEA
     ORANGEORANGEORANGEORANGEORANGEORANGE
     ^                                  ^

(b)  LISPLISPLISPLISPLI
     PASCALPASCALPASCAL
     ^           ^
```

FIG.2

KEY 1 : AB
KEY 2 : CDE
} ACBDEACB……………

在選擇不同的 KEY 時，要注意的是所有 KEY 的用度是不能有公因數（COMMON FACTOR），否則所產生出的KEY 就會短了很多。

圖二（A）就是兩個沒有公因數的KEY 的混合，產生出來的長度是 30。

圖二（B）是兩個有公因數 2 的 KEY，新的 KEY 的長度是 12。

如果我們用的三個 KEY 是：IBM，APPLE，COMPUTER，新的KEY 的長度便等於3×5×8＝120。

程式的列印見圖三。

行 50—230 是給用者輸入本文檔案、密碼檔案的名稱，KEY 的數目和各個 KEY 的名稱。

行 270—520 是每個 RECORD 的讀、譯和寫的迴圈。

行 340—490 是譯每一行文字的迴圈。

行 540—650 是再次印出用者輸入的資料，以便作記錄。

CP／M的 I／O 構造爲我們帶來了一些問題。例如CP／M 是用 CTRL—Z 作爲一個檔案結尾的標記。萬一我們一個文字譯成 CTRL—Z 時，並把 它寫在密碼文件上。當我們以後讀取密碼文件遇到這個 CTRL—Z 時，文件以後的資料便無法讀取。同樣地，RETURN 也會替LINE INPUT帶來了錯誤。因此，行 460便是用來檢查出所有 NUL，LF，RETURN 和 CTRL—Z等文字，用原來的文字代替之。

使用的方法見圖四。選擇KEY 的時候要特別留意各

Plaintext:	H	e	l	l	o
Plaintext (ASCII):	01001000	01100101	01101100	01101100	01101111
Key:	10010101	11000100	00110101	11011010	01010110
Ciphertext:	11011101	10100001	01011001	10110110	00111001

FIG 1

```
10 ' FILE ENCRYPTION PROGRAM USING VERNAM'S ALGORITHM WITH ANY NUMBER OF KEYS
20 '
30 ' NO COPYRIGHT IN THIS PROGRAM, ANY PART OF IT CAN BE RE-PRODUCED IN ANY
40 ' FORMAT WITHOUT THE PERMISSION OF THE AUTHOR

50 ON ERROR GOTO 660
60 HOME
70 DEFINT A-Z
80 PRINT TAB(32);"FILE ENCRYPTION":PRINT
90 ' *** GET FILENAMES AND KEYS ***
100 VTAB 3:INPUT "INPUT FILE ";INF$
110 OPEN "I",#1,INF$
120 VTAB 5:INPUT "OUTPUT FILE ";OF$
130 OPEN "O",#2,OF$
140 VTAB 7:INPUT "NUMBER OF KEYS ";N
150 PRINT
160 IF N>10 THEN PRINT CHR$(7):GOTO 140
170 DIM K(N),L(N),K$(N),T$(N)
180 FOR I=1 TO N
190  PRINT "KEY";I;": ";
200  INPUT "",K$(I)
210  K(I)=1
220  L(I)=LEN(K$(I))
230 NEXT I

240 ON ERROR GOTO 0
250 HOME
260 PRINT "Working";
270 ' *** LOOP ON RECORDS ***
280 FOR I=1 TO 32767
290  IF EOF(1) THEN 530
300  LINE INPUT #1,IN$
310  L=L+1
320  LENGTH=LEN(IN$)
330  OT$=""
340 ' *** LOOP ON CHARACTERS WITHIN RECORDS ***
350  IF LENGTH=0 THEN 510
360  FOR J=1 TO LENGTH
370   Z$=MID$(IN$,J,1)
380   P=ASC(Z$)
390   FOR K=1 TO N
400    T$(K)=MID$(K$(K),K(K),1)
410    P=P XOR ASC(T$(K))
420    K(K)=K(K)+1
430    IF K(K)>L(K) THEN K(K)=1
440   NEXT K
450   Y$=CHR$(P)
460   IF P<>0 AND P<>10 AND P<>13 AND P<>26 THEN 480
470   Y$=Z$
480   OT$=OT$+Y$
490  NEXT J
500  PRINT ".";
510  PRINT #2,OT$
520 NEXT I
530 CLOSE

540 PRINT " Done!"
550 PRINT:PRINT "PLAINTEXT:";TAB(19);INF$
560 PRINT:PRINT "CIPHERTEXT:";TAB(19);OF$
570 PRINT:PRINT "LINES: ";TAB(18);L
580 PRINT:PRINT:PRINT "NUMBER OF KEYS: ";TAB(18);N
590 PRINT
600 FOR I=1 TO N
610  PRINT "KEY";I;": ";K$(I)
620 NEXT I
630 VTAB 22:PRINT
640 PRINT "Please remember these information and Goodbye!"
650 END

660 ' ERROR CHECKING ROUTINE
670 IF (ERR=64 OR ERR=53) AND ERL=110 THEN RESUME 100
680 IF ERR=64 AND ERL=130 THEN RESUME 120
690 IF (ERR=6 OR ERR=13) AND ERL=140 THEN RESUME
700 ON ERROR GOTO 0
```

MBASIC 譯碼/解碼儀

KEY 的長度，像圖四便是一個不大好的例子：因為Z－80 CARD 和 HANDKER CHIEF的長度分別是9和12，有一個公因數3，使結合出來的KEY 長度減短了。所以所有 KEY 的長度最好是質數。

另外，KEY 的意思也是很重要，最佳的 KEY 是沒有意思的文字，如Q×％$＋P或＊：＝∧NM 等，但缺點是這些文字不容易記憶，需要寫在紙上，因而容易遺忘。如果用名字，國家或電話號碼等的 KEY ，它的好處是方便記憶，但是會很容易給敵方悉破。

圖五是程序完結時顯示出用者輸入的資料。由於程序是用 MBASIC 寫成，所以速度頗慢。圖六便是本文檔案和密碼文件的比較，這個譯碼程序約花了三十五秒。如果要提高速度，最好選用 MBASIC COMPILER ，我想速度應該可以由三十五秒提升到五秒以內。用 COMPILER 之前，請把行170 的N改為 10。

這個程序得出來的密碼雖然不是波法破解，但是如果使用適當長度的KEY 加以保護的話，密碼專家也會認為這程序為他帶來了不少麻煩。

備註：因為WORDSTAR DOCUMENT FILE 的WORD 結構特殊，所以本程式並不適用。■

```
                                        FILE ENCRYPTION

INPUT FILE ? PLAIN.TXT
OUTPUT FILE ? CIPHER.TXT
NUMBER OF KEYS ?  3

KEY 1 : BALLOON
KEY 2 : Z-80 CARD
KEY 3 : HANDKERCHIEF
```

FIG.4

```
Working..... Done!

PLAINTEXT:              PLAIN.TXT

CIPHERTEXT:             CIPHER.TXT

LINES:                  6


NUMBER OF KEYS:     3

KEY 1 : BALLOON
KEY 2 : Z-80 CARD
KEY 3 : HANDKERCHIEF

Please remember these information and Goodbye!
```

FIG.5

```
A>TYPE PLAIN.TXT
This is a very great secret:
  The way to decrypt the ciphertext is just only to run the encryption
program again and uses the ciphertext as the input file and the same
keys.

  Please don't tell any body, thank you!
```

FIG.6

```
A>
A>TYPE CIPHER.TXT
▄ESK▄ .sRTEVo#'(&U◤HA*791j
▄◣zG!g.!* NT▄,432&WE▌W$,u-.PZ][=#$0!NBF 15!⊏1NUB▄O=}2) I■NH$i
3 %^HE]#)3
8-EV_M(j7)?JW H);}6/CO)/v,#_TPX>")?rKA Y!,{'3SMB ,62&}KQP 6"2c:CQS
A.2!h

r          b@D+7>m9MZSj(;.=            SZWb)4$=♠BC&&4f (FF_
```

WORDSTAR 高級應用篇：

蘋果長老

MAILMERGE 的操作備忘

續完

當我們利用WORDSTAR把所有的變數插入後，執行一MAILMERGE 的時候，它會自動替你安排段落，使符合你要求的規格，這時和文字處理系統中編輯功能的重定命令（CTRL—B）功能一樣。有時MAILMERGE 的重新安排與用家意願發生衝突時，我們可以利用一些點命令來改善，使達到我們的要求。

以下我將印制表格時點命令的操作概括：

爲了要加入點命令，我們要按CTRL—N ，在結尾按（RET），但控制格式的點命令，須在.PF ON的狀態下才有效，不過在印表時點命令的確完成了多方面的功能，例如報表的規格印制可以用.PF來控制，.PF有著三種模式：

① .PF DIS（這是標準狀況）

這個命令會使MAILMERGE 自行作決定，當它碰到變數時便會起動報表印制規格，一直到CARRIAGE RETURN出現才停止。

② .PF OFF：

這個命令會促使MAILMERGE 完全不採用報表印制規格，即使由於變數的插入而改變行的長度。

③ .PF ON

這個命令使MAILMERGE 起動報表印制規格，直至.PF OFF或.PF DIS 出現才停止，利用輸入的規格安排報表格式，若果MAILMERGE 再碰到其它點命令（以下列出）的話，則會依照點命令的規定安排。

MAILMERGE 其實是利用點命令來操作的程式，在控制邊界時可以利用.RM DIS （標準狀況）來達成，如果你要把報表的右邊界定在第n格的話，可以利用.RMn 來做到。其次在很多時我們都希望把輸入對齊，在這方面MAIL—MERGE 提供了三個命令：

①.IJ ON 一當報表印制時，這個命令會使輸入對齊，在.RM DIS 狀態下跟著右邊界變動，在.OJ DIS 狀態下，MAILMERGE會自動把輸出調整。

②.IJ OFF 一當報表印制時，若果右邊參差不齊，MAIL—MERGE 在.RM DIS 狀態下會挑出最大的右邊界爲標準，而在.IJ DIS狀態下則輸出不整齊的格式。

③.IJ DIS 一輸出和輸入完全一樣。

若果閣下希望利用以上的命令來處理書信及表格的話，你要注意在檔案之中下一個空白出現之前的長度，若超過設定的邊界，則不印在同一行，而從下一行印起，同時在下一個空白出現之前，若果碰到雙引號括起的文字，假如空白之後的字長超過邊界，而印到邊界時，你先把字拆開分成兩行來印。

磁碟的更換及報表印制的控制：

MAILMERGE 可以在執行插入檔案和使用資料檔案前，給使用者換磁碟或改變使用的磁碟機，只要在.FI和.DF命令中的檔案名稱之後增加"CHANGE" 即可。至於報表印制的控制方面可以分爲三類：

① 有條件的頁次分割——若果在檔案之中有n行不能印在同一頁上的話，可以利用.CP n命令，使這n行印在下一頁之上。

② 無條件的頁次分割——.PA 命令可以令MAILMERGE 在遇見這命令時由下一行開始分割。

③ 頁次的控制——MAILMERGE 可以接受三種控制模式

a .PN n可以設定頁次而且立即印出。

b .PC n可以設定頁次的欄位（標準是33格）。

c .OP 不印出頁次。

除了這些外，MAILMERGE 尙提供用家自定頁次的空間、表頭和註脚的調整，使文件得以盡善盡美。

在頁次空間控制方面，MAILMERGE 提供了三個命令，它們是：

①.PL n—用來控制每頁印出的行數（標準爲66行）。

②.MT n 一用來控制上邊界（標準爲3行）。

③.MB n 一用來控制下邊界（標準爲8行）。

至於控制表頭和註脚方面，MAILMERGE 亦提供四個命令予使用者，它們是：

①.HE TEXT 表頭（即表格之開頭，它出現在每一頁的第一行，注意TEXT的長度不超過一行）。

②.FO TEXT 註脚（取代頁次，出現在每頁的最後一行，TEXT的長度同樣不超過一行）。

③.HM n—這個命令用來控制表頭邊界（表頭和文章間的空行，標準是兩行）。

④.FMn 一這個命令用來控制註脚邊界（文章和註脚間的空行，標準是兩行）。

寫到這裏我要將 MAILMERGE 的操作備忘結束了，下次再會。■

SCREEN SAVER

相信各位都有許多遊戲軟件，其中有些必然玩得好精。有沒有想過在玩到遊戲最難一關的時候，將該幀畫面保留起來以便向朋友炫耀？

假如遊戲是沒有加上保護，要截取這有紀念價值的畫面會相當容易，可惜一般的遊戲軟件都是加上保護，要在中途停止遊戲來截取高解像圖就較爲麻煩。雖然市面有WILDCARD帮手，但步驟繁複，會使人的興緻大減。

那些有保護的遊戲，大部份都是在控制DOS 方面做了手脚，遊戲一旦執行後，便不能隨意控制主機。假如中途按CTRL RESET來中止遊戲，則大部份的情況下會立即RE-BOOT 磁碟機，或是HANG 機。假如是前者，由現在起，閣下可以很輕鬆的便可取得這些有紀念價值的圖畫了，那些WILDCARD大可置之高閣。

當然，如果你熟識機械語言，這個程式不會難倒你。

現在讓我解釋如何使用這個程式。首先放入SYSTEM MASTER 入磁碟機，BOOT起DOS 3.3。然後打入程式。放入一張全新的空磁碟入磁碟機，打入INIT HELLO，按RETURN。磁碟在INIT後，便會有一個名HELLO 的程式了。

當我們要保留某遊戲的任何一幀畫面時，只要放入這張有〔HELLO〕程式的磁碟，便可輕易取得。這個程式的主索引有6個選擇，包括：

(1) 觀看第一頁高解像
(2) 觀看第二頁高解像
(3) SAVE第一頁高解像
(4) SAVE第二頁高解像
(5) CATALOG磁碟
(6) EXIT程式

相信有了這個程式後，你可將各項的輝煌遊戲記錄向朋友炫耀了！

文末所列出的程式可以讓你們輕易截取任何的高解像遊戲畫面。在未講解程式用法之前，待我先解釋其中的竅妙。

那些有保護的遊戲，大多數在CTRL RESET 方面做了手脚，只要按下，電腦立即會REBOOT機或HANG 機。對於後者，本程式是無能爲力，由於必須關機，那麼高解像頁裏的東西會被全部洗掉。筆者設計這個程式主要是針對前者，利用REBOOT 機這個過程，而毋須使用WILDCARD。

按動RESET會REBOOT磁碟機，假如在REBOOT 機之前放入另一張磁碟，則BOOT 時會將該張新機上的HELLO載入記憶體，而不影响到高解像第一頁裏的內容。假如這個HELLO 程式是我這個程式，那麼我們便可以輕易再取回主機的縱控權，該幀圖畫就如探囊取物。當然假如那個遊戲在設計時，如果按下RESET會重新BOOT 機並將第一頁的畫面劃花，則本程式就無法可施了。幸好這類的軟件不算太多，市面仍有半數軟件合這程式使用。■

```
5   REM ******************
8   REM **              **
10  REM ** SCREEN SAVER **
20  REM **      BY      **
25  REM **  COMPUHOUSE  **
30  REM ** COPYRIGHT(C) **
35  REM **     1985     **
38  REM **              **
40  REM ******************
100  TEXT : HOME : NORMAL
105 T =  - 16303: REM  TEXT
110 G =  - 16304: REM  GRAPHICS
115 P(1) =  - 16300: REM  PAGE 1
120 P(2) =  - 16299: REM  PAGE 2
125 A(1) = 2000: REM  HEX PAGE 1
130 A(2) = 4000: REM  HEX PAGE 2
135 KB =  - 16368: REM  KEYBOARD
140 Z =  - 16336: REM  SPEAKER
145  POKE  - 16297,0: REM  HI-RES
150  POKE  - 16302,0: REM  FULL
     PAGE
197  REM
198  REM     TITLE
199  REM
200  INVERSE
205  FOR K = 1 TO 5
210  VTAB K: HTAB 14
215  PRINT  SPC( 14): NEXT
220  VTAB 3: HTAB 15
225  SPEED= 100
230  PRINT "SCREEN SAVER"
235  SPEED= 255: NORMAL
240  POKE 34,6: REM  LOWER TOP
297  REM
298  REM     MAIN MENU
299  REM
300  HOME
305  PRINT "      1.SHOW PAGE 1": PRINT
310  PRINT "      2.SHOW PAGE 2": PRINT
315  PRINT "      3.SAVE PAGE 1": PRINT
320  PRINT "      4.SAVE PAGE 2": PRINT
325  PRINT "      5.CATALOG DISK": PRINT
```

Screen Saver

```
330  PRINT "   6.EXIT PROGRAM"
335  VTAB 20: HTAB 20
340  POKE KB,0: GET K$:K =  ASC (
     K$)
345  IF K < 49 OR K > 54 THEN 340

350 K = K - 48: PRINT
355  ON K GOSUB 400,400,500,500,3
     75,390
360  GOTO 300
375  GOSUB 600
380  PRINT  CHR$ (4)"CATALOG": PRINT

385  HTAB 20: GET K$: GOSUB 600: RETURN
390  TEXT : GOSUB 600: END
397  REM
398  REM     SHOW SCREEN
399  REM
400  GOSUB 600
405  POKE P(K),0: POKE G,0
410  POKE KB,0: GET K$
415  IF K$ = "1" THEN K = 1: GOTO
     405
420  IF K$ = "2" THEN K = 2: GOTO
     405
425  POKE T,0: POKE P(1),0
430  RETURN
497  REM
498  REM     SAVE SCREEN
499  REM
500  INPUT "NAME OF SCREEN:  ";K$
505  IF K$ = "" THEN  RETURN
510  PRINT  CHR$ (4)"BSAVE";K$;",
     A$";
515  PRINT A(K - 2);",L$2000"
520  GOSUB 600
525  RETURN
597  REM
598  REM     SCREEN WIPE
599  REM
600  FOR W = 1 TO 40
605  POKE 33,W
610  HOME :X =  PEEK (Z) +  PEEK
     (Z) * 256
615  NEXT : RETURN
```

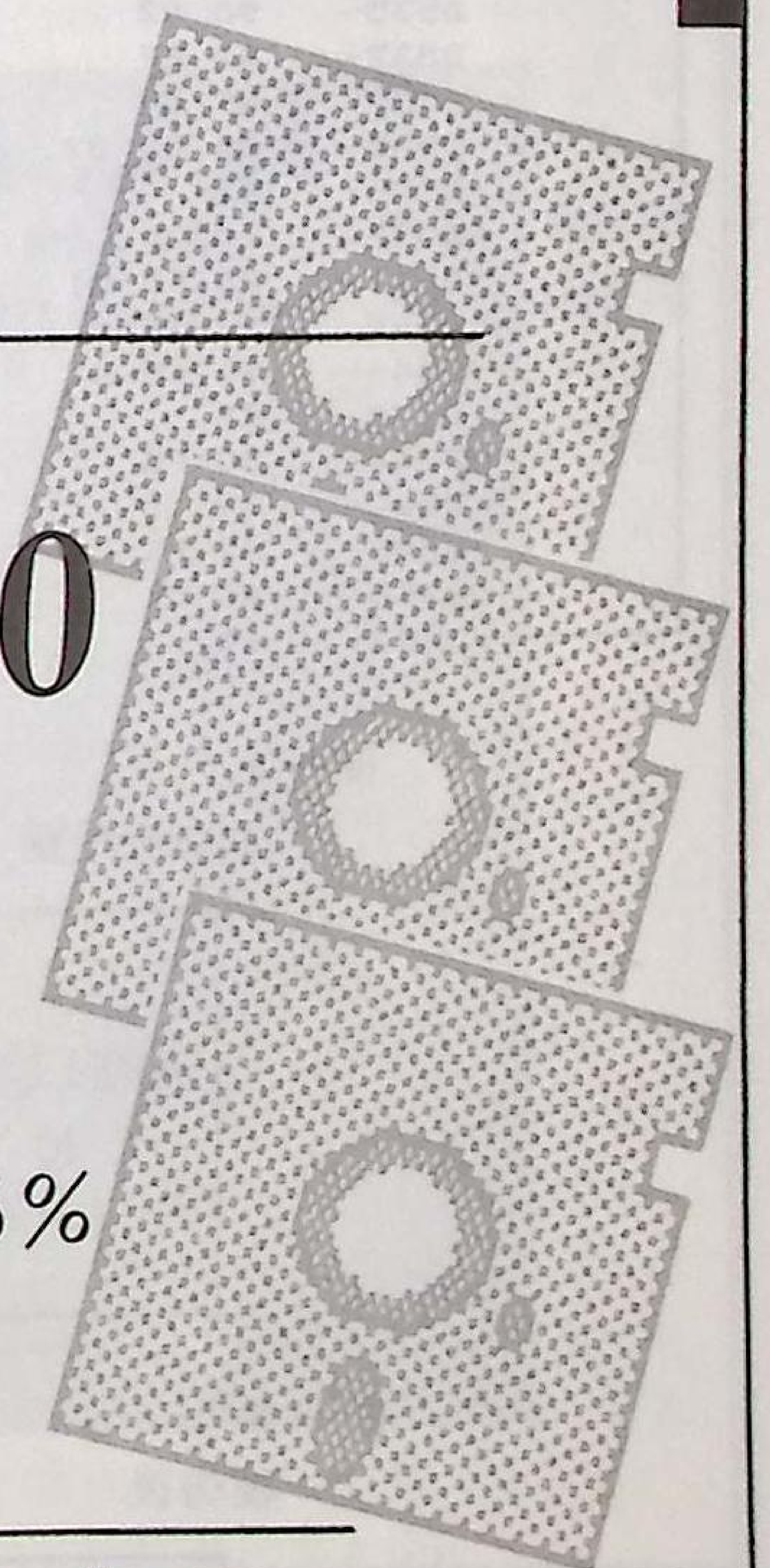

COMPUHOUSE 最新出品軟件

CP/M-FORMAT 40

套裝軟件

大喜訊

CP/M用戶可將磁碟容量提高15%



每套附精裝MENU只售50元

特約經銷處：

電腦時代讀者服務部

百老滙銅鑼灣分店（三越百貨公司對面）

百老滙旺角分店（凱聲戲院隔鄰）

COMPUHOUSE出品 電腦時代出版社總代理

在機械語言程式中CALL BASIC

陳家麟

有位朋友對我說：「BASIC CALL 機械碼就易啫，但倒轉由機械程式CALL BASIC 就難了。唉！簡直不可能……」

是否眞的不可能？如果要做到這點，首先我們要了解一BASIC 的操作及讀取我們輸入的命令之實際情況。爲什麼我們打入RUN〔RETURN〕後，電腦會知道我們叫它做什麼呢？

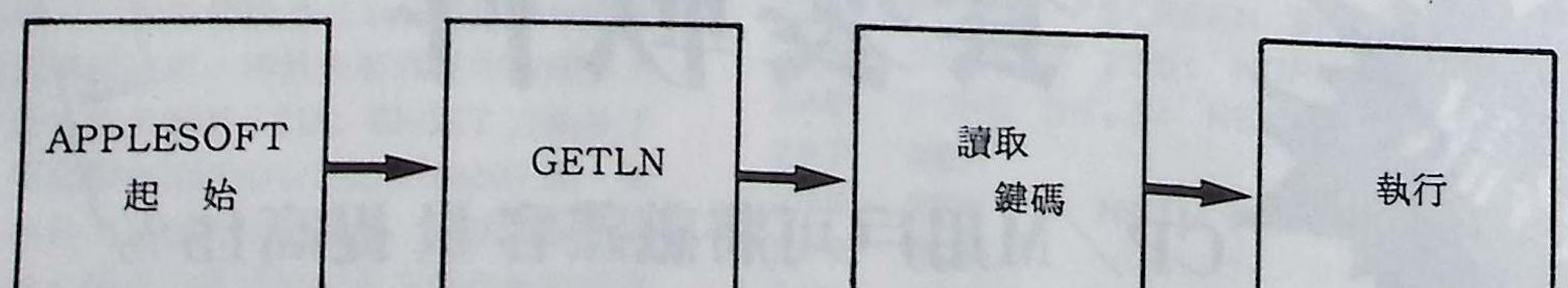
圖二

```
        *D43CL

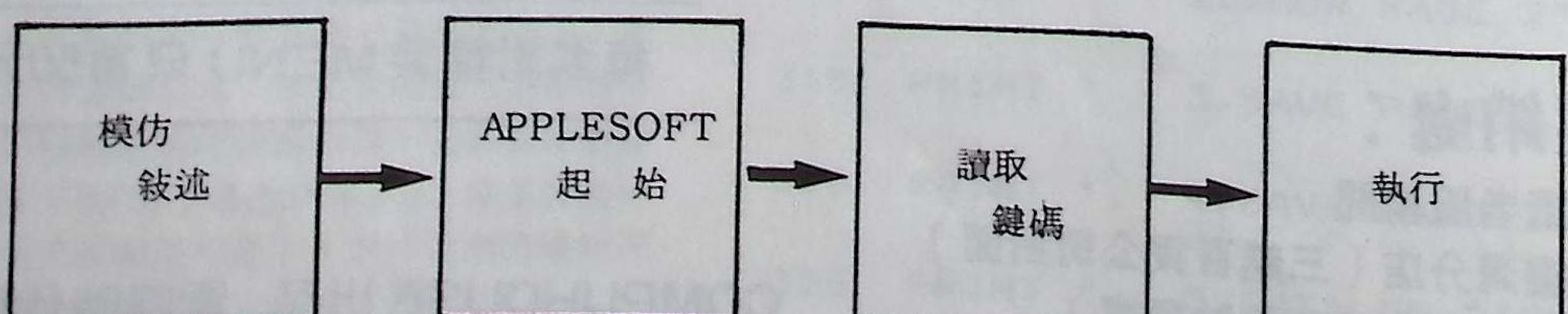
        D43C-    20 FB DA    JSR    $DAFB
        D43F-    A2 DD       LDX    #$DD
*  ──── D441-    20 2E D5    JSR    $D52E
   ───► D444-    86 B8       STX    $B8
        D446-    84 B9       STY    $B9

   ───► D52E-    86 33       STX    $33
        D530-    20 6A FD    JSR    $FD6A ──► GETLN
        D533-    E0 EF       CPX    #$EF
        D535-    90 02       BCC    $D539
        D537-    A2 EF       LDX    #$EF
        D539-    A9 00       LDA    #$00
        D53B-    9D 00 02    STA    $0200 , X
        D53E-    8A          TXA
        D53F-    F0 0B       BEQ    $D54C
        D541-    BD FF 01    LDA    $01FF , X
        D544-    29 7F       AND    #$7F
        D546-    9D FF 01    STA    $01FF , X
        D549-    CA          DEX
        D54A-    D0 F5       BNE    $D541
        D54C-    A9 00       LDA    #$00
        D54E-    A2 FF       LDX    #$FF
        D550-    A0 01       LDY    #$01
   ──── D552-    60          RTS
```

圖一a.
本來流程圖



圖一b
模仿後



其實，APPLESOFT 本身是機械碼寫成，這點是大家知道的。我們試打入APPLESOFT 的起始地址：E003，LIST出來看：

〝E003— 4C 3C D4 JMP $D43C” （附圖二）

再LIST $D43C，又是一個JSR，一個又一個的JS—R。如果一直看下去，可以看到一個JSR$D52E，再LIST $D52E 便可看到JSR$FD6A。咦！咁熟口面嘅？原來它就是GETLN 副程式，卽是說APPLESOFT 做到這裏便等待一行敍述，然後依敍述工作。試想：若我們可刪掉JSR$FD6A，而預先模仿一行敍述（用機械碼），那便成了。（附圖一）

只要把敍述的ASCII碼存入$200.2FF，再於敍述後加〝：”卽$BA，因爲APPLESOFT 讀到有〝：”才分析敍述。

例：〝RUN：” ASCII碼則爲：$D2，$D5，$CE和$BA。

程式則爲：

```
LDA  #$D2
STA  $200
LDA  #$D5
STA  $201
LDA  #$CE
STA  $202
LDA  #$BA
STA  $203
RTS
```

當然不一定用RUN，閣下還可用GOSUB或GOTO等敍述。

在機械語言程式中CALL BASIC

當然ROM 內的記憶體是不容易改變，剛才說把JSR＄FD6Ａ刪掉，則並非沒有可能。由於APPLESOFT 起始到這指令間不過是幾個BYTE，所以我們只須“抄”其中幾個BYTE再呼叫一些副程式，把這些指令集起成爲一條短程式便成功了。

還有一件事：GETLN 後回來的指令是ＣＰＸ，所以G—ETLN最後一指令必須存在，所以再加一個LDY＄35 便可。（程式見圖三）請比較原程式。

現在的用家喜歡寫ＢASIC＋機械式PROGRAMME，對於他們來說，這程式應有不少方便之處。還有，它不受絕對地址影响，即是放在那裏也很方便，毋須改變程式中JMP的地址。

請不要少看這短短的“抄”程式，其背後所花的時間和心血已不少了。各位朋友，只要我們懷着信心和耐心，無論在研習電腦上甚至在做人處事方面，任何沒可能的事都可以做得到。■

圖三

```
0300-   A9 D2       LDA   #$D2   ┐
0302-   8D 00 02    STA   $0200  │
0305-   A9 D5       LDA   #$D5   │
0307-   8D 01 02    STA   $0201  ├ 模仿鈸述
030A-   A9 CE       LDA   #$CE   │
030C-   8D 02 02    STA   $0202  │
030F-   A9 BA       LDA   #$BA   │
0311-   8D 03 02    STA   $0203  ┘
0314-   20 FB DA    JSR   $DAFB  ┐
0317-   A2 DD       LDX   #$DD   ├ 抄起始
0319-   86 33       STX   $33    ┘
031B-   A4 35       LDY   $35    - GETLN最後命令
031D-   20 33 D5    JSR   $D533  ┐
0320-   4C 44 D4    JMP   $D444  ┘ 執行
0323-   00          BRK
```

將APPLESOFT程式轉爲TEXT檔案來編輯

你有沒有想過將一個BASIC 程式轉成爲一個TEXT FILE ，然後用一些文字處理器（如APPLEWRITER，PIE WRITER等）來進行讀取、編輯與偵錯。在某些情況下，這是非常實用的。

當閣下嘗試如此做時，可將下列幾句的程式加入你要編輯的程式的開首。然後RUN 該程式便可。程式會產生一個名〔TFILE〕的TEXT FILE 將整個程式轉爲TEXT 形式。留意在RUN 程式之前，査看磁碟上有沒有其他名爲TFILE 的檔案，新檔案會將舊檔案掩蓋掉。

```
5  REM   ***********************
10  REM   ** TEXT FILE MAKER **
15  REM   ***********************
20    D$ = CHR$ (4)
   : REM   CTRL D
25    PRINT D$;"MON C,O"
30    PRINT D$;"OPEN TFILE"
40    PRINT D$;"WRITE TFILE"
50    POKE 33,30
70    LIST 100 -
80    PRINT D$;"CLOSE"
90    TEXT
   : END
100   REM  TEST OF THE PROGRAM
110   REM  CA#14
120   REM  MODIFIED VERSION
130   REM  BY TAM YIN
```

要讀取這個TEXT FILE 方法很簡單，只要用下列的程式即可：

```
10 D$ =  CHR$ (4)
20  PRINT D$;"OPEN TFILE"
30  PRINT D$;"READ TFILE"
40  GET A$: PRINT
50  PRINT A$;: GOTO 40
```


讀者服務部搬遷

電腦 時代讀者服務部 經已由舊址
廣東道582號廣發商業大厦搬遷到
九龍上海街438－444號同珍商業中心

蘋果奧妙



張峯寧

本文主要的目的，是根據個人經驗，將一些鮮爲人知的蘋果電腦奧秘，盡量提出，共同研究、發掘。

(一) 如何測試出磁碟上BINARY FILE PROGRAM 即（B型FILE）之起始及終止點呢？

其實要解決這個問題很簡單，只要首先將程式BLOAD入記憶系統，而此時程式之起始位址存在$AA72、$AA73，而其長度則存於$AA60、$AA61，至於其結束位址即起始位址加上長度之值。只要各位在監督程式下，找出此地址的值，便可！

(二) 如何測試電腦最高可使用的地址？

位址$73、$74（即十進制之115，116）是存放APPLESOFT最高可使用之地址。如果電腦爲48K的話，一般其值是$BFFF。所以我們可以更改115，116之位址，以減少程式可使用之空間。因此，我們可以檢查115，116之值以確知最高可用位址。

(三) 爲何GR後RESET會出現許多反白的@字呢？

原因是因爲我們將同樣一個資料存入同樣的記憶位址，但却處於螢幕設定方式不同之情況（GR&TEXT），故RESET後看到的是黑白的文字形與彩色的方塊的混合畫面。因爲它們有相同的對應編碼。這同時也指出一個頗有趣的問題——GR的空白（即黑色）與TEXT之空白（即黑色）是對應着不同的編碼。（文字幕一個字相當於相疊的兩個方塊）

(四) 如何加快磁碟機存取資料時速度？

由於一般磁碟機平常不存取資料時，馬達是靜止的，而當有命令要磁碟機工作時，馬達才打開，但當時馬達轉速不穩，故需一段時間延遲再去存取資料，以致讀寫速度緩慢。

要解決此難題，除了可用RAM DISK之方法外，我們可在程式中直接啓動馬達，而增加讀寫的速度。當控制咭插於SLOT6時，觸動49385（$C0E9）位址可啓動馬達，而49384（$C0E8）則可關掉馬達。例如：我們可在直接方式下鍵入？PEEK（49385），此時磁碟機之馬達會轉動，但游標一樣會出現，這時如我們RUN／LOAD／SAVE一個程式時，速度是會比平時快的!!

(五) 文字幕地址的計算方法：

表面上，APPLEⅡ供文字幕使用的地址是由$0400～$07FF（即1024個地址〔1K〕），但事實上可看出最底下8列均有8個地址並未應對到SCREEN上。（它們是分配給8個SLOT使用的）

至於如何計算文字幕地址的方法很簡單。首先我們要把SCREEN看成座標圖，橫的爲X，直的爲Y，只要將欲印出的位置（X，Y）值代入下列的公式即可：

A) INTEGER BASIC: A＝1024＋X＋(Y/8)×40＋(YMOD8)×128

B) APPLESOFT BASIC:

M8＝Y－INT(Y/8)×8 —— (1)

A＝1024＋X＋INT(Y/8)×40＋M8×128 —— (2)

A＝ADDRESS

如果各位是用APPLESOFT的話，本人提供了一個簡短的程式，以供讀者作運算之用。（程式一）。各位只要輸入（X，Y）之值便可找出相對的地址了。

(六) 如何在印字機上顯示"70行之顯示畫面"？

現在市面上有一種程式，可在一部普通主機上（不需任何擴展卡）顯示70行（COLUMN）。（程式名稱：FLEX TEXT）

而由於此程式是以作圖方式繪出的（即用程式來控制螢幕的輸出），因而我們可用圖形COPY的技巧將整個畫面轉印到列表機上。其操作過程與印圖形完全一樣，首先打開印字機（PR#1），再指定所要COPY畫面的頁數（POKE 1913，2），再把控制碼17送出（PRINT CHR$(17)），這樣即可。

```
10  REM  *** PROGRAM OF CALCULATI
     OG ***
20  REM  ***         POKE ADDRESS
     ***
30  REM  ***            BY
     ***
35  REM  ***          TONG CHANG
     ***
40  HOME : INPUT "ENTER X VALUE:"
     ;A
50  INPUT "ENTER Y VALUE:";B
60 M8 = B -  INT (B / 8) * 8
65 C = 1024 + A +  INT (B / 8) *
    40 + M8 * 128
70  HTAB 7: VTAB 5: PRINT "THE AD
    DRESS IS:";C;
80  HTAB 5: VTAB 7: INPUT "MORE C
    ALCULATING (Y/N)?";C$
90  IF C$ = "Y" THEN 40
100  HOME : PRINT "GOODBYE...!!":
     END
8080  HTAB 5: VTAB 7: INPUT "MORE
     C ALCULATION(Y/N)?";C$
```

RWTS程式內部剖析

莫信

RWTS不僅讀寫資料，還能設定磁片啓始狀態，更重要的是在錯誤發生時能復原，若能夠好好加以利用，它比RUN、LOAD等的指令還要實用，要使用此程式，使用者要先產生一輸入輸出區（簡稱IOB），及磁碟機設備特性表（簡稱DCT）。

在DOS手冊中，RWTS的進入點是3D9，而3D9則將指標指向B7B5，所以B7B5才是RWTS程式啓始處，但眞正是RWTS程式位於BD00處，B7B5的作用只是把中斷關閉，並呼叫眞正的RWTS程式，待眞正RWTS程式囘返之後，再把中斷功能重新恢復，並在進位旗號中傳囘RWTS的囘返碼。

其實DOS內部也有IOB及DCT，位址B7E8爲輸入輸出區（IOB），而位址B7F8則爲磁機特性表（DCT）。而IOB是由最後一次DOS動作設定，你可以使用它，但使用自已的IOB比較完全穩妥。

RWTS主程式在BD00，它首先把輸入輸出區位址儲存在48及49，在BD08處將槽編號乘16置於X暫存器內，並與先前一次輸入輸出區的值相比，若值比變了（BD19爲舊值），便等至由磁碟機上傳來的8個相同字元找到爲止，這使前一槽的磁碟機停止動作，避免負荷過重的現象。

BD2D程式選定新槽號位址，並儲存下來以備使用。BD3F程式把磁碟機打開。在BD44的程式則把磁碟機特性表的位址及緩衝區位址分別存在位址3C、3D、3E、3F，因而提取資料時比較容易。BF50處的程式把輸入輸出區中的磁碟機編號取出與先前的相比較，若不同則更新資料。

BD5F處選擇磁碟機編號（1或2），並把磁碟機特性表中第2及第3位元組存到46及47，作爲控制步進馬達計時之用。在BD75處，取讀磁軌號碼，並呼叫磁軌搜尋程式。BD7D處的程式則作磁碟機特性表計時之用，在BD8A讀取指令碼，作相對應的行動（若指令碼爲0，由BDE5執行NOP指令；若值爲1，便執行BD9B讀入程式；若值爲2，便執行B800副程式；若值爲4，程式轉移至BDE7，設定初值）。

B86A程式控制磁碟機介面，使用到的位址稱爲裝置選擇位址（DEVICE SELECT ADDRESSES）。B—AIE爲DOS 系統控制讀寫磁頭前進後退的程式，而BA59則打開步進馬達，並JSR BA7F 延遲時間，延遲時間由DCT決定。步進馬達由BA6A處關閉。

在BE3B的程式爲找尋磁軌的程式，由DCT求得相位計數，並呼叫BE4C程式，將所有馬達關閉，最後跳至BA1E步進程式。

當完成工作後，BE27處的程式會淸除進位旗號，並返至呼叫RWTS的程式，亦即是控制RWTS副程式。

在此，我要提醒各位，若你們使用CALL RWTS作讀寫工作後，再進入監督程式後想囘返到BASIC時，你的APPLE便無法再正確的執行命令（例如使用3D0G），因監督程式使用"GO"指令時，APPLE會將位址48的內容存入P暫存器中，而上文亦提到RWTS利用位址48及49存入IOB的位址，問題便發生了，惟一解決的方法是每次完成JSR RWTS 後將0存入位址48，如此，你便可以自由進出監督程式。■

武士道KARATEKA



解解拆拆

武士道是本月最受歡迎的遊戲軟件，它不但容易使用，而且很有故事性，人物的造型和動作亦非常生動，因此到高登商場抄這個遊戲的人很多。但由於武士道並非一般軟件的可以由COPYA來抄，因此坊間出了二個版本，一個是原裝版只有一面，另一個是WILDCARD 版，共有兩面四個LEV—EL，可以由COPYA來抄，但效果極之差，不但沒有故事，而且要手動才可以上LEVEL 。有見及此，本人特將武士道的抄法心得公開，但各位讀者不必爲了BACKUP 而担心。

設備：

A）APPLⅡ+及兩部磁碟機。

B）EDDⅢ或LOCKSMITH5.0。

C）原裝武士道及空碟一張。

抄法：

A）先調校兩部機至相近速度。較快的做DRIVE 1，較慢的做DRIVE 2。

B）先COPY＄0至＄1磁軌。

C）再COPY＄2.25的磁軌。

D）最後COPY＄4至＄22的磁軌。

E）複製成功。

小型製作篇之四

如何 BOOT DRIVE 2

BABEL

你有沒有遇到過，當你使用某些軟件，在設定好程式碟在DRIVE 1，儲資料的碟DRIVE 2 後，才發覺調轉了放；又或者在COPY 完磁碟之後，要BOOT的碟却在DRIVE 2；又或者擁在一個能行40 TRACK 磁碟機，及一個只能行35 TRACK 的厚身磁碟機的朋友，在用F1D抄FILE 時，設定了SOURCE DRIVE及DESINATION DRIVE 之後，才發覺設定錯了呢？所以，最好就是DRIVE 1與DRIVE 2 能夠隨時調換，而且可以在DRIVE 2 BOOT碟，就更加好了。

最近筆者在高登，看見有代人改裝DISK DRIVE控制咭的服務，但改裝收費絕不便宜。而筆者也曾研究過APPLE DISK DRIVE 控制咭的綫路，發覺只需十分簡單的修改，便可以隨時調轉DRIVE 1和DRIVE 2 的位置；可惜不知道該電腦店，是否用筆者的方法改，如果有讀者知道的話，請來信告訴筆者。

原理概述

讀者如果擁有磁碟機操作手冊的話，可參閱該手冊中所載的控制咭綫路圖，其中一部份如圖(一)所示。圖中的ENABLE 1及ENABLE 2 之分別是DRIVE 1及DRIVE 2 的選擇綫，當BOOT碟或是在DRIVE 1 模式下，ENABLE 1 便會選擇它所連接的DRIVE 1，在DRIVE 2 模式下，ENABLE 2 便選擇它所連接的DRIVE 2，而不同的模式是由使用者所輸入的命令中的D所決定（D1或D2)。換句話說，

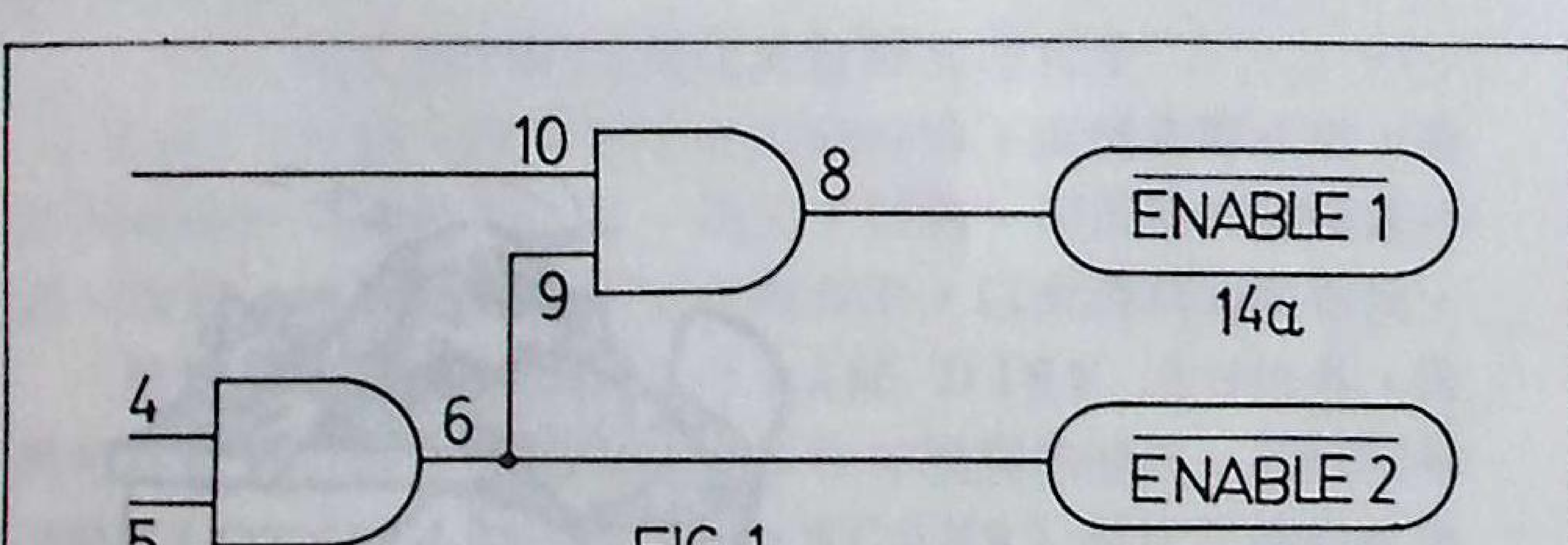


FIG 1

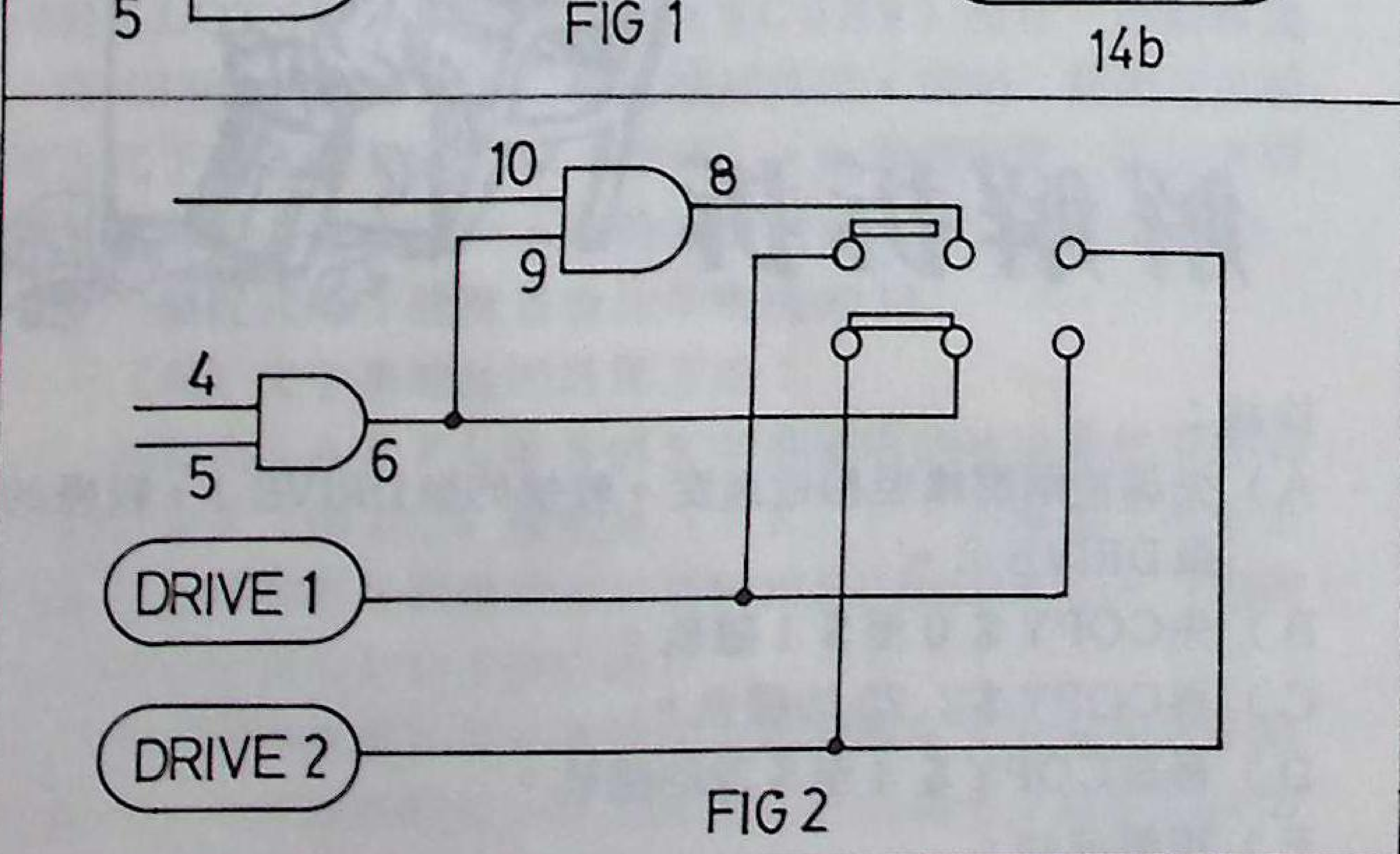
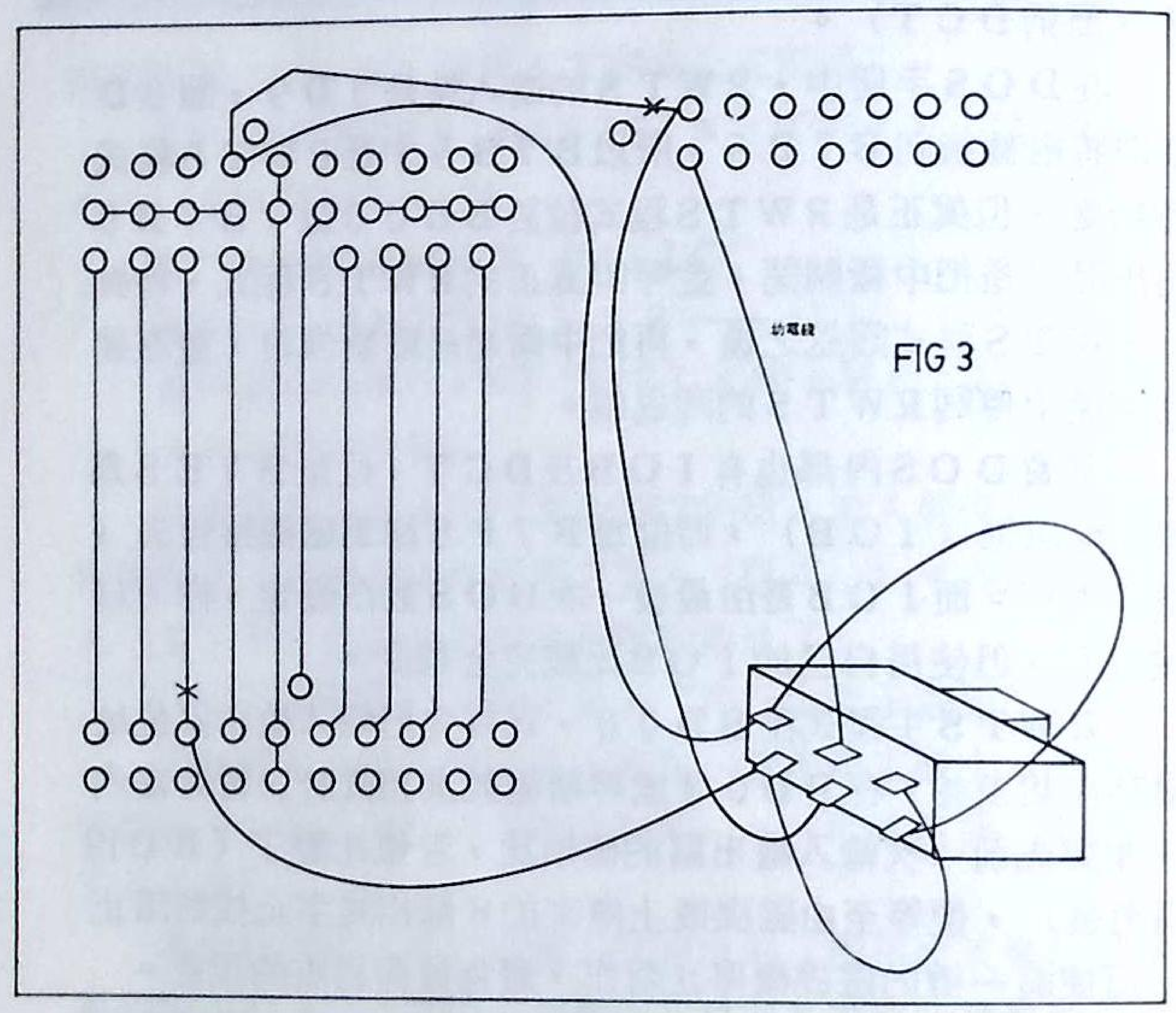


FIG 2



我們就可以利用這兩條選擇綫的特性，來達到我們的要求，在正常狀態下，命令中的D1就選擇DRIVE 1，D2 就選擇DRIVE 2，但在另一狀態下，這兩條綫便對調了接DRIVE 1及DRIVE 2，當命令中用D1或BOOT碟時，却選擇了DRIVE 2，D2則選擇了DRIVE 1。總之DRIVE 1和DRIVE 2 在這種狀態下，像是完全對調了。我們祇需要在控制咭加上一個雙刀雙擲的開關，便可以在任何情況下，都可以輕易選擇兩況狀態的任何一種。我們需要作的綫路修改見圖(二)。

操作簡述

讀者可從圖(一)中，看出兩條選擇綫分別來自74LS132的第6脚及第8脚，而這兩條選擇綫分別接往插座1及插座2的第14脚，我們便是要把這兩條綫與插座之間割開，把它們接往開關作選擇，然後再由開關接回插座1及插座2，我們便利用這雙刀雙擲開關作NORMAL MODE及INVERSE MODE的選擇。

圖(三)所見是控制咭的部份底板圖。圖中有X的是綫路需要割開的地方，並指示出如何接綫往開關。我們在焊接引接之前，先要固定開關在底板。在控制咭的零件面，近左上角處有一條甚濶的綫路，我們就是要把開關焊接在這條綫路上，所以先是刮去一些綠油，面積比開關稍大一些，當然開關要越細越好。圖(四)所示是如何用焊錫來固定開關在底板上。圖(五)是完成圖，在改裝完成後，需要在開關上貼上一塊絕緣膠布，防止因碰撞而令其他介面咭短路。

後記

這個磁碟機控制咭的改裝工作，只需一個小型開關及小量幼電綫，費用比上一篇改裝16K RAM CARD 更便宜。希望讀者會滿意。如有任何關於硬件的製作或改裝的提議，歡迎向筆者來信提出。■

各位如不想自己動手改動 CONTROLER 咭，讀者服務部可提供此項服務。持讀者優待證的讀者可獲折扣優待。

任何詢問，請電：3-312007。

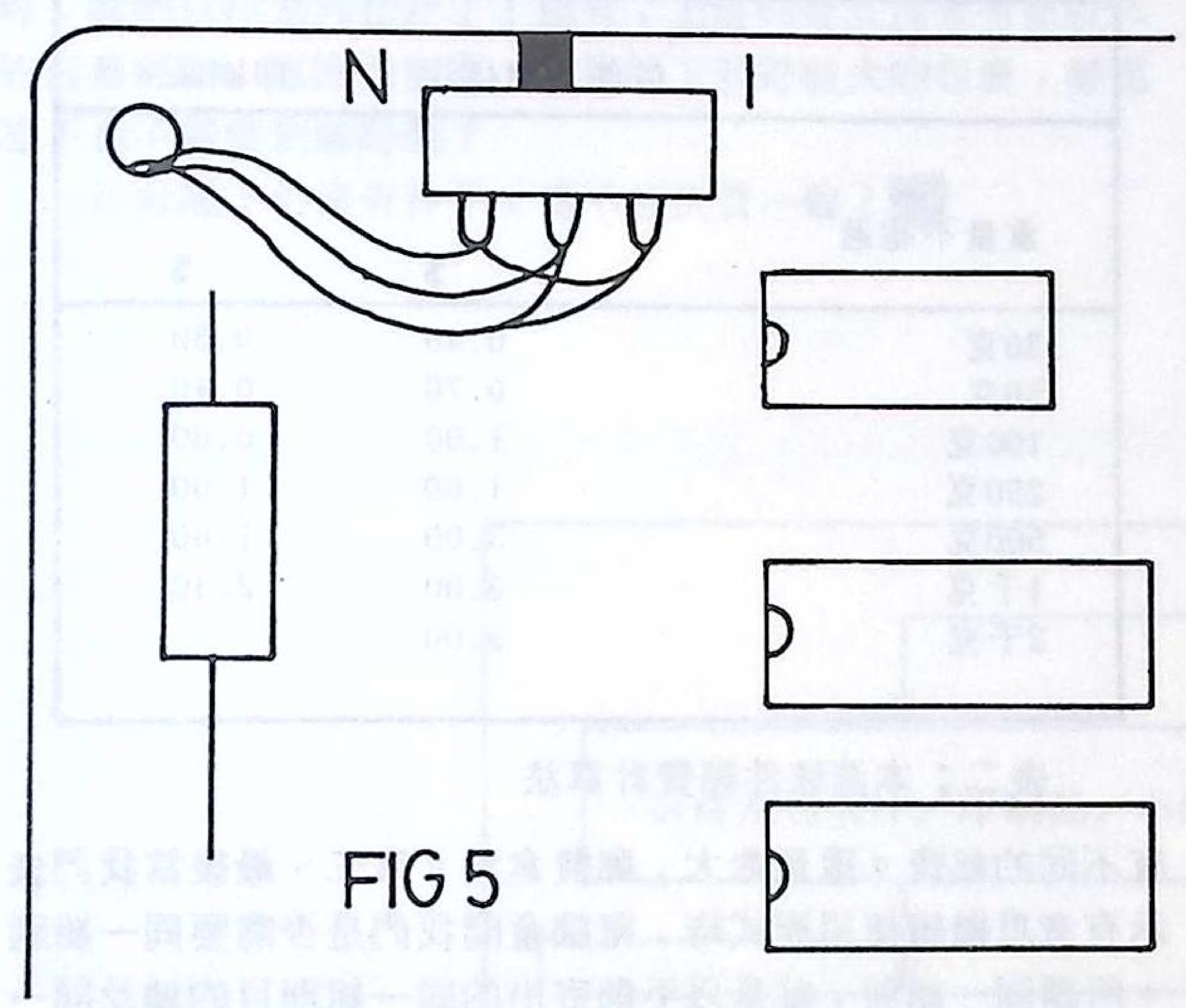


FIG 5

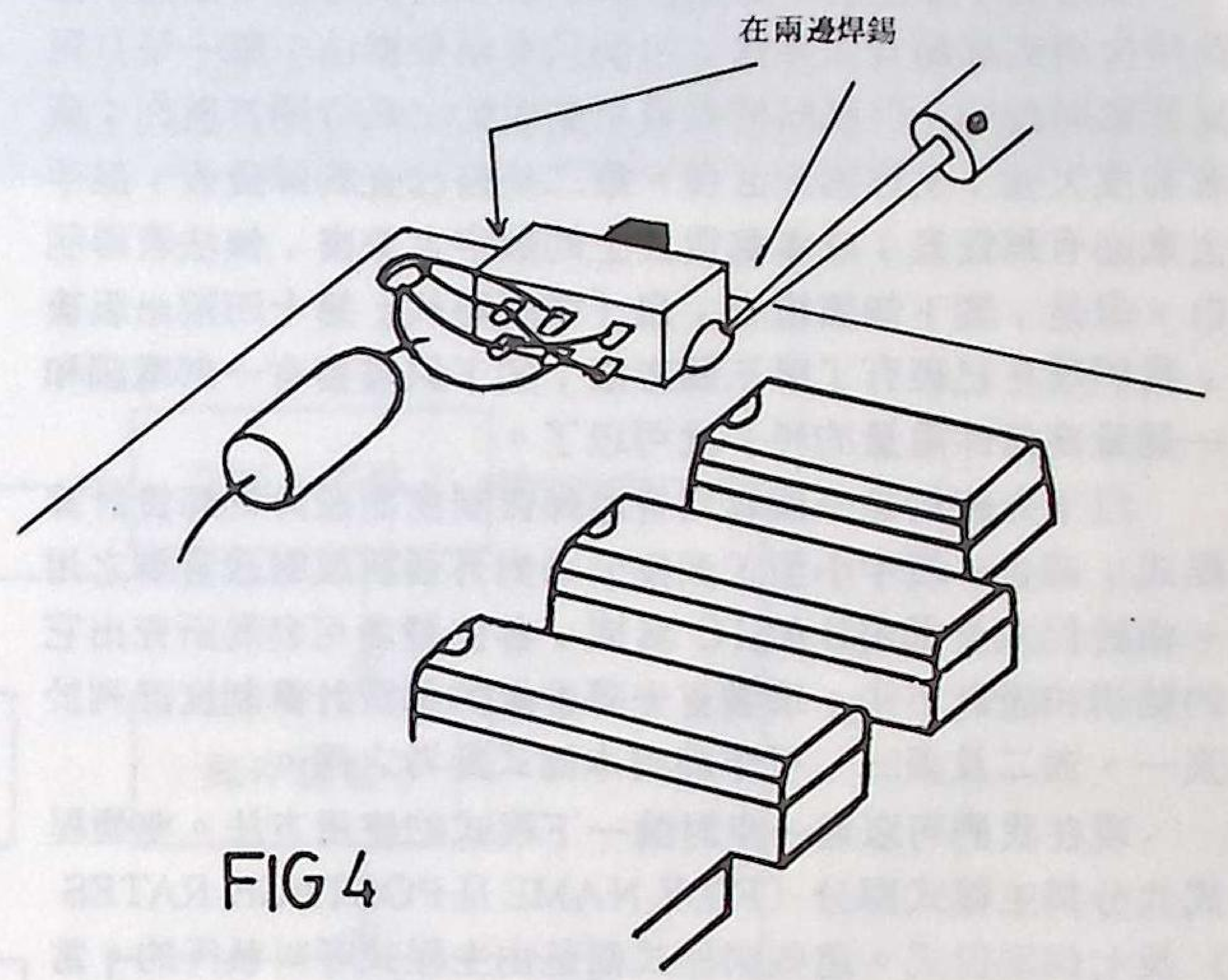


FIG 4

令CP/M用戶獲取額外15%的磁碟容量！

適合44K、56K、60K系統使用

FORMAT 40

現時絕大部份的磁碟機都是40TRACK 並融的產品，爲了這個原故電腦時代不遺餘力爲各位提供了一系列的DOS 40 TRACK 軟件及文稿，使到經濟效率提高15%，而且在運作上更爲流暢，工作效率改進了30%。

長期以來，得到最強大軟件系統支持的CP／M 系統由於資料較少，在磁碟格式化後僅能使用35 TRACK ，加上大型軟件系統往往都佔用超過100K BYTE ，使工作空間餘下很少，若果僅有一部磁碟機的話，相信都很難享受CP／M 的強大軟件好處了。

當然現時已有80 TRACK 的磁碟機，不過售價仍很貴，加上很多人原本已擁有了磁碟機，這樣對花上二千元來買80 TRACK 的磁碟機似乎不合化算，有鑒於此，電腦時代推出了FORMAT —40軟體系統，解決以上問題的矛盾。

FORMAT —40可以令任何CP／M 版本增加磁碟貯存量15%，包括44K ，56K及60K三種，而且買家在不久將來更可以以更相宜價格購買配合128K～256K RAM DISK 的40 TRACK CP／M 系統。對於只擁有單磁碟機而希望運行CP／M 的朋友，FORMAT— 40是你必須要買的軟體！■

適合中小型工商業寄郵件及貨辦用途的
香港郵資計算諮詢系統

何炳根

假若閣下手上有一封信件或小郵包需要寄出，而又不想費時失事到郵局査詢郵費，以前只有兩個辦法：第一是打電話到郵局詢問，但郵局的職員可能太忙，無時間答覆你；或者態度欠佳，使你感到沮喪。第二是自己査閱郵費表，但手上未必有郵費表；或者郵費表上的數字太繁複，無法看得明白。但是，閣下無須担心，自「電腦時代」第十四期出版後，我們現在已經有了第三個方法，閣下只需要有一部電腦和一個量度信件重量的秤子就可以了。

以下介紹的是一個專爲香港郵費制度而設計的郵費計算程式，適合一般中小型工商業公司對外通訊或寄送貨辦之用。由於程式全部由BASIC寫成，各位讀者可容易研究出它的結構和設計方法。筆者首先將香港的郵費計算制度詳列於表一、表二及表三，可作爲與本程式參考之用。

現在我們可以進一步討論一下程式的使用方法。整個程式共分爲主程式部分（FILE NAME是POSTAGE RATES）及七個副程式。這些副程式都是由主程式呼叫執行的。當RUN 主程式後，電腦會詢問一連串的問題，我們需要逐一回答。爲了讓讀者清楚起見，筆者將這些問題撮要成表四。

		一區	二區
信件及明信片	首10克	$1.00	$1.30
	每加10克	$0.60	$0.70
*二等郵件	首10克	$0.60	$0.90
	每加10克	$0.40	$0.50

*印刷品及小郵包均屬此類，是不能封密以妨碍檢查的

表一：空郵郵費計算法

使用方法

主程式部分主要是計算空郵的郵費。當選擇空郵及輸入郵件的重量後，我們跟著需要輸入郵件的目的地的區域號碼。除了大部份的亞洲區國家或地方是屬於第一區外，其它全部國家都是屬於第二區的。輸入區域號碼後，電腦會繼續詢問郵件的種類及是否需要掛號寄出（REGISTRATION）。通常無論是空郵或平郵，掛號郵件比普通郵件是需要多付港幣4元郵費的。電腦計算出郵費後，當然希望知道閣下是否繼續使用這個程式。

假若我們選擇的是平郵，則首先需要回答郵件的種類及目的地，主程式此時會按我們所輸入的類別呼叫各有關副程式。當電腦找出適當的副程式後，我們再按次序回答各問題。有幾點是值得一提的：第一、當郵件的種類是小郵包時，電腦就不會再詢問郵件的目的地，因爲各地的小郵包郵費都是相同的，讀者可參考表三。第二、當電腦進入副程式後，螢幕會顯示一系列重量界限，以表示郵件在不同的界限內會

重量不超過	信件及明信片 $	印刷品 $
30克	0.40	0.30
50克	0.70	0.40
100克	1.00	0.60
250克	1.60	1.00
500克	3.00	1.60
1千克	6.00	2.40
2千克	9.00	

表二：本港郵件郵費計算法

有不同的郵費。重量愈大，郵費愈貴。第三、最後當我們表示有意思繼續使用程式時，電腦會問我們是否需要同一組別。所謂同一組別，就是以平郵寄出的同一組別目的地及同一種類的郵件。這樣，在同一組別時，電腦就可以在同一副程式內進行運作。假若我們需要的是不同組別，則程式會自動轉入主程式部分再進行運作了。

日後修改

當然，郵費並不是固定不變的。想當年一毫子寄一封信，眞是唏噓不已。爲了使程式適應所謂時代的進步，我們少不免在郵局調整郵費時需要改動程式內的資料及計算部分。主程式的資料及計算部分是在行號1020、1030、2020及2030。行號1020及1030是計算一區空郵的郵費；而行號2020及2030是計算二區空郵的郵費，大家可參考表一就可以知道應該改變那幾個數字來配合新的郵費。至於平郵方面，每個副程式的資料部分都在行號150及200，大家亦可參考表二及表

	信件及明信片		印刷品		小郵包*
重量不超過	中國澳門台灣 $	其他國家 $	中國澳門台灣 $	其他國家 $	$
20克	0.50	1.00	0.40	0.90	2.00
50克	1.00	1.50	0.60	1.30	2.00
100克	1.90	2.40	1.00	2.00	2.00
250克	4.30	4.80	1.80	3.60	4.00
500克	8.40	9.20	3.00	6.50	7.00
1千克	14.40	16.00	5.40	10.80	12.00
2千克	20.40	26.00	7.20	15.00	

表三：本港以外地區平郵郵費計算法

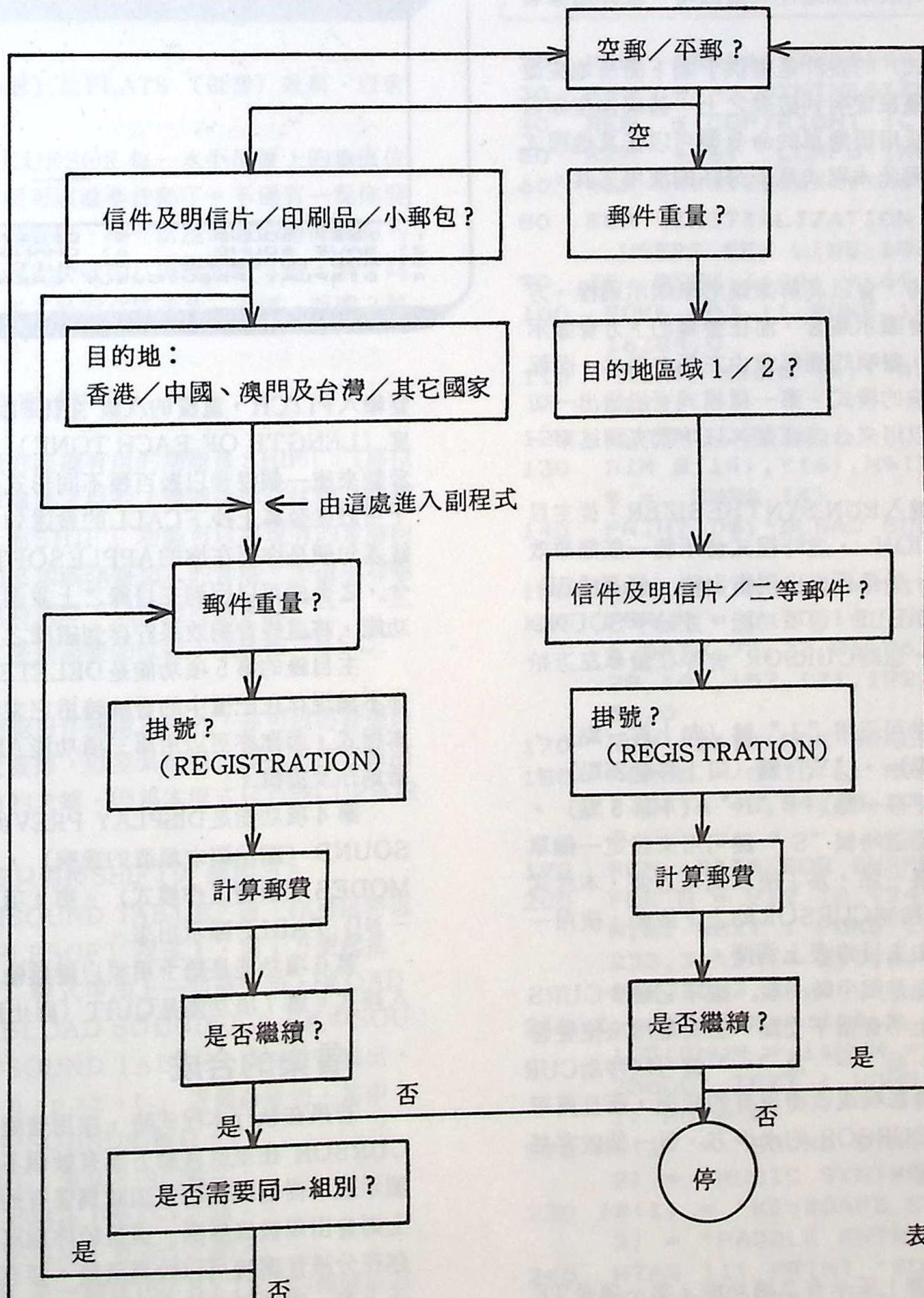
*所有海外目的地均設有小郵包服務，重量以一千克爲限

香港郵資計算諮詢系統

三。不妨自己試試暫時充當郵政局，隨意將郵費加多十倍八倍，然後嘗試修改程式內的資料及計算部分，看看是否正確。不過，有一點是需要注意的，就是這種改變程式的方法只適宜純粹是郵費的調整，假若當整個計算郵費的方法都改變時，我們只好另外設計了！還有，上述的程式沒有考慮較大的包裹郵費，原因很簡單。若果閣下要寄較大的包裹，難道閣下還不需要到郵局嗎？

假若閣下仍沒有秤子，還不趕快買一個？■

由於本文程式列表太長之故，編輯部不欲浪費篇幅，本程式大家可於本期的程式磁碟#14B內找到。



表(4)

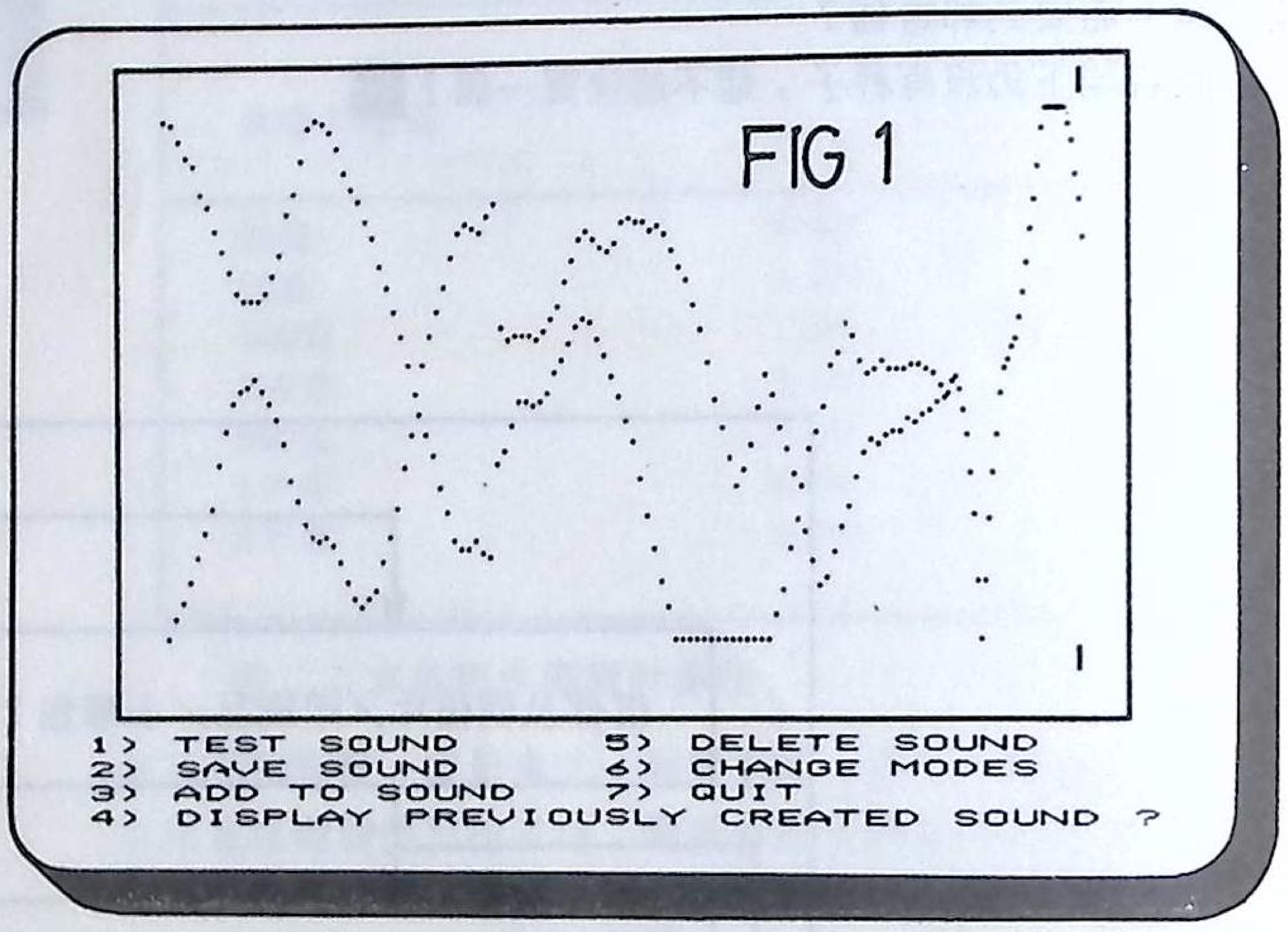
能隨意製造任何聲音效果加插入自己的程式之中

用法簡易，效果奇佳

以軟體來達成的聲音合成器

各位朋友相信都知道APPLE 是可以創製出不同效果的聲响，不過若果以APPLESOFT 來編寫這類型的程式時，相信最多只能做出一些簡單的聲調而已。雖然坊間新近推出有MOCKING BOARD，但筆者相信並非每個人都擁有。假如你想將音樂或其它音响效果加進程式裏面時，就非用筆者這個軟件不可了。

聲音合成器（本程式）的設計是提供予閣下簡易地製造及貯存一些包括音樂，機械音响到磁碟之上，甚或加在你自己的程式之中，你只要採用很簡單的命令便可以在其他程式中提存你創造的聲响，因此本程式是十分好的應用工具。



使用本程式

本程式在製造音响時，會以高解像圖形來顯示過程，方便參考。在螢幕之上方會顯示高音，而在螢幕的下方會顯示低音。當你重播音响時，聲响的圖形會由左至右顯示，而程式本身提供兩種創造音响的模式，第一種模式可以造出一切的音响效果，第二種可以用來合成音樂。以下我先講述第一種模式。

使用本程式時可以鍵入RUN SYNTHESIZER，從主目錄上先選擇INSTRUCTION ，這時程式會示範一些簡單效果，當這個階段結束後，螢幕再次出現指引時，便選擇BE—GIN NEW SOUND TABLE 這項功能，並給予SOUND TABLE 一個檔案名稱，這時CURSOR 會停在螢幕左方給你創造聲音。

控制CURSOR 移動可以用〝I〞鍵（向上移一點），〝U〞鍵（向上移動5點），〝Y〞鍵（向上移動25點）。同時向下可用〝M〞（下移一點），〝N〞（下移5點），〝B〞（下移25點），而這時候〝S〞鍵可用來設定一個單音並將CURSOR 向右移一點，爲了使用上更方便，本程式加插了用 JOYSTICK 來控制CURSOR 的上下移動，使用一JOYSTICK 功能可以由主目錄表上選擇。

SPACE BAR的功能是與平時一般，按下它會令CURS—OR 向右移而在螢幕上不會留下光點，主要是用來使聲音在重播時產生暫停效果，而〝→〞及〝←〞鍵可以移動CUR—SOR 向右或左而不會影响或改變現有的音調，而且更可以利用數目字鍵來移動CURSOR 的水平方，每一個數字都對應不同的螢幕水平位置。

主目錄的功能

本程式的主目錄如圖1所示有7項功能，第一項是TE—ST SOUND ，此項功能可以用來聽你所設計的聲音，只要輸入PITCH，重覆的次數（REPETITIONS）及音調的長度（LENGTH OF EACH TONE），你可以調校以上這些參數來使一個聲音以數百種不同形式出現，當你滿意的時候，可以從螢幕上抄下CALL的敍述句（如圖2所示），這個敍述句便是你要在你的APPLESOFT 程式之中所加入的命令，之後便可以回到主目錄之上並選用SAVE SOUND 這項功能，將這些音响效果貯存到磁碟之上。

主目錄的第5項功能是DELETE SOUND ，這功能是你不滿現存在記憶中的音响時用它來清除記憶體並重新開始本程式，而你亦可以用第三項功能ADD TO SOUND 來重新創作及編輯。

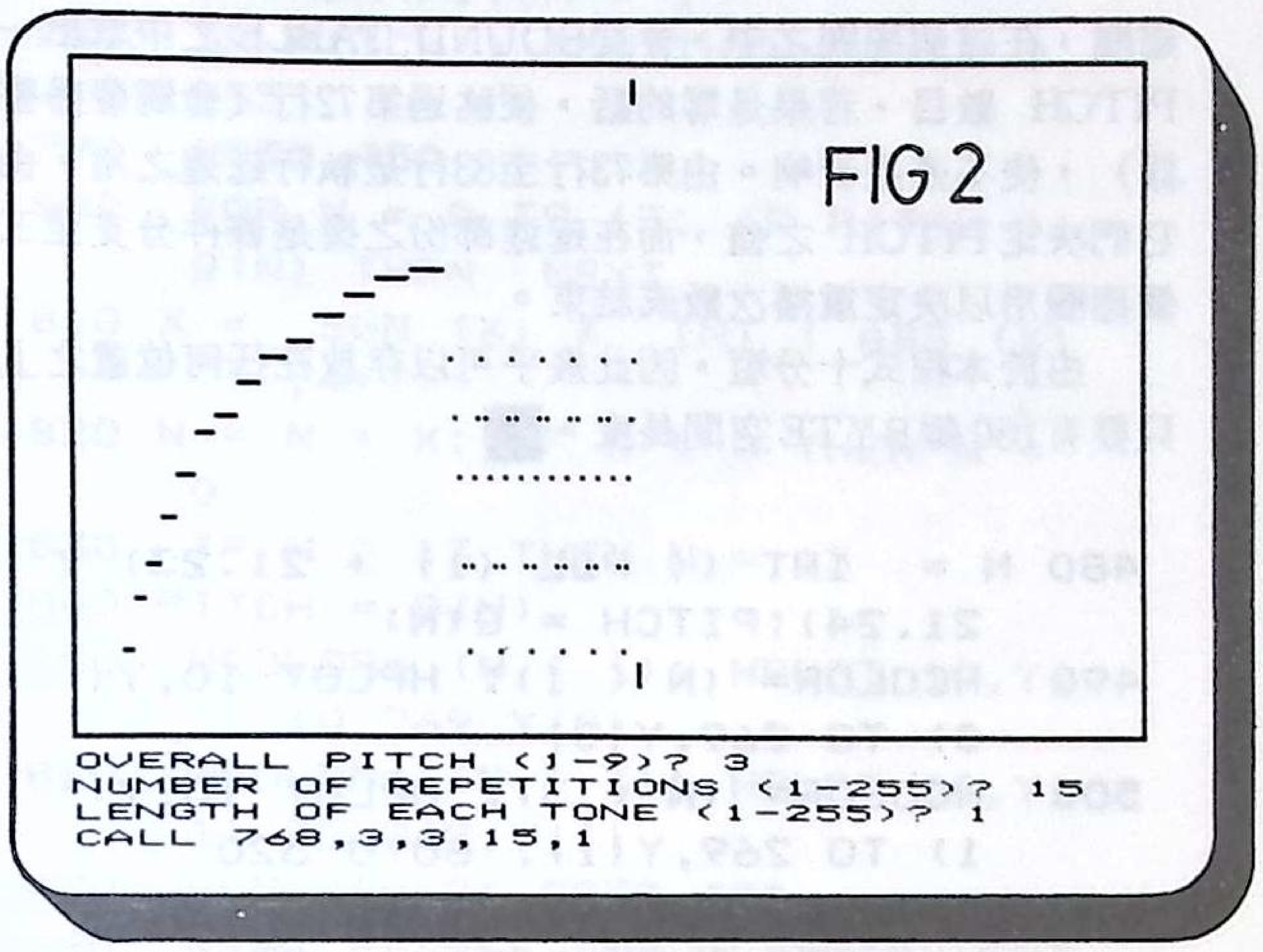
第4項功能是DISPLAY PREVIOUSLY CREATED SOUND（顯示剛才創造的音响），及第6項CHANGE MODES（改變工作模式）。第4項功能可以將你的SOU—ND TABLE顯示出來。

第6項功能是給予用家以鍵盤輸入或以JOYSTICK 輸入模式。第7項功能是QUIT（終止程式）。

音樂的合成

若果在執行本程式時，採用音樂合成模式，你會發現了CURSOR 在垂直移動方面有數個不同之處，主要是如圖3顯示般，從全域性的音調減爲僅有的14個特定音符。而螢幕上則會出現數條橫線，這是給你編寫音樂時用的輔助線，使你在分辨音調時可以較爲清楚。要有效地編寫音樂，你需要手上有一些樂譜，注意在輔助線之間是代表音調，這時你會發覺CURSOR 的垂直位置會相對應這些音調，不過本模式

以軟體來達成的聲音合成器



並不包括SHARPS（高音）及FLATS（低音）效果，以求使用上的簡易。

你可以跟著樂譜將CURSOR 每一水平位置上的垂直位置來相對輸入，這樣你便可以重奏音樂了。不過有一點你要緊記，高音會比低音的速度快，換句話說高音的時間較短。在實際上，最高的C調會比最低的D調時間短3倍，爲了同步起見，你可以把高音奏多數次來抵消此一毛病，而圖3將各音調的重覆次數列出。

SHIFT 的特別功能

另一項在音樂合成方面很有用的功能是SHIFT （移位），這個功能是用來作補遺之目的，舉例說你在試聽之際溜了一些音在開始位置，而SHIFT 功能可以將現有的音階向右移使騰出足夠的空間來加添你要加入的音階。而這項功能可以在編寫及編輯音樂時選擇，只要使用數字及“→”“←”鍵即可將CURSOR 移到需要的位置，然後按“ESC”鍵即可選用此項功能。

除了將SHIFT功能用作補遺之用外，亦可以將音樂向左移使多餘的音調被覆蓋掉，但要緊記做完這項工作之後需將CURSOR 移回音樂的末端，因爲本程式只合成CURSOR左邊的訊息。

要離開SHIFT功能只需按SHIFT 0 鍵即可。

在創製及貯存一個SOUND TABLE之後，你可以將這些聲響應用到你的APPLESOFT 程式上，爲了方便起見，你只要在APPLESOFT 程式中加上一句命令句，BLOAD SOUND. READER及BLOAD SOUNDFILE，其中SOUNDFILE是你創製的SOUND TABLE，要令聲音輸出，只要加上CALL 768，n，p，r，l， 這個命令句，其中：

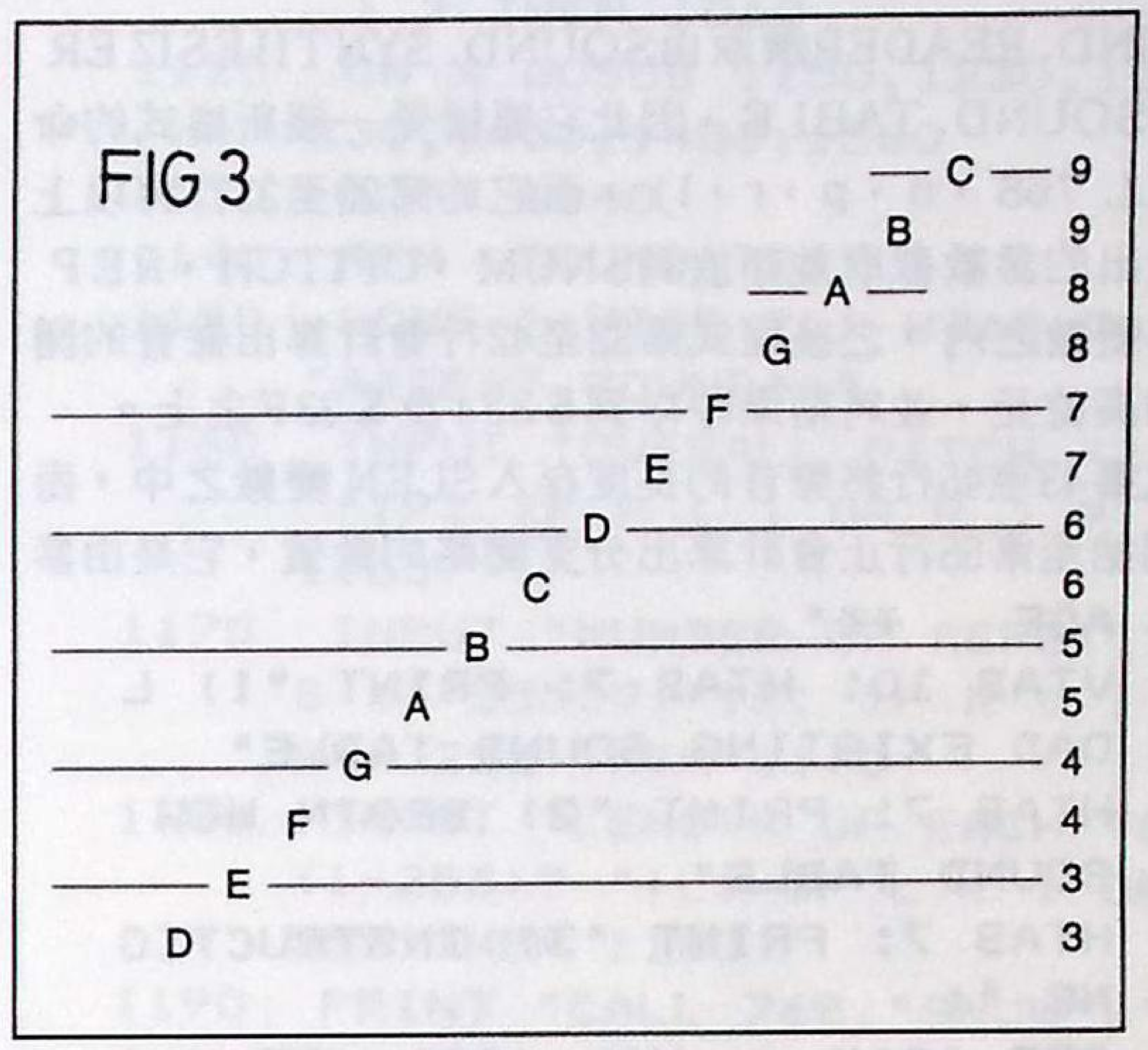
n 是SOUND TABLE 中SOUND的數目。

p 是音調的間隔，必須在1至9之內的數字。

r 是音響重播的次數，必須在1至255之間。

l 是音調維持的長度，必須在1至255之間。

SOUND. READER 是一個僅160 BYTE 的機械語言程式，它的原程式如列表1所示，它主要是呼叫$C030 位




```
20  REM  ***********************
30  REM  *    SYNTHESIZER      *
40  REM  * COPYRIGHT (C) 1985 *
50  REM  * BY   COMPUTING  AGE *
60  REM  ***********************
80  REM  INITIALIZATION ** PRODOS
     USERS SEE LINE 1860
90  IF  PEEK (104) = 64 THEN 120
100  POKE 103,1: POKE 104,64: POKE
     16384,0
110  PRINT  CHR$ (4);"RUN SYNTHES
     IZER"
120  LOMEM: 37376
130  DIM Q(14),Y(6),M$(2),I$(2):D
     $ =  CHR$ (4)
140  PRINT D$;"BLOAD SOUND.READER
     "
150  REM  DATA FOR MUSICAL NOTES
160  FOR N = 0 TO 14: READ Q(N): NEXT
     : DATA 73,77,86,96,108,116,1
     28,144,152,171,192,213,230,2
     54,0
170  REM  DATA FOR SCALE LINES
180  FOR N = 0 TO 6: READ Y(N): NEXT
     : DATA 48,54,65,77,89,107,12
     8
190  REM  DATA FOR SHAPE TABLE
200  FOR N = 912 TO 924: READ A: POKE
     N,A: NEXT : POKE 232,144: POKE
     233,3: DATA 2,0,6,0,8,0,4,0,
     172,30,7,32,0
210 Z = 12.5:M = 1:I = 1:PITCH =
     125:SNUM = 1:NUM = 1:STADD =
     25600: TEXT : HOME : SCALE=
     1: ROT= 0
220 M$(1) = "SOUND SYNTHESIS":M$(
     2) = "MUSIC SYNTHESIS"
230 I$(1) = "KEYBOARD ENTRY ":I$(
     2) = "PADDLE ENTRY    "
240  HTAB 11: PRINT "SOUND SYNTHE
     SIZER": VTAB 22: PRINT "** C
     OPYRIGHT 1984 BY  COMPUTING
```

址（內置ROM 的發聲副程式進入點），這時加上適當的參數及延遲網路可令APPLE 發出任何音響。

SOUND. READER讀取由SOUND. SYNTHESIZER所產生的SOUND. TABLE，因此它要接受一個新格式的命令（CALL 768，n，p，r，l），而它的第20至31行將以上命令所發出的參數截取並存放到SNUM，OPITCH，REP及TLEN變數之內。之後程式第32至42行會計算出聲音的開始位址的高位元，並將結果貯存到＄324及＄33F之上。

程式第43至45行將聲音的長度存入SLEN 變數之中，而第46行開始至第55行止會計算出分支網路的數值，它是由參數P所決定，然後第85行會產生一個條件分支至75至83其中一行，而75行會產生最大的延遲。第56至96行是產生聲音的迴圈，在這個迴圈之中，會從SOUND TABLE之中取出一PITCH 數目，若果是零的話，便跳過第72行（會觸發揚聲器），使不產生音响。由第73行至83行是執行延遲之用，由它們決定PITCH 之值，而在延遲部份之後是條件分支至三個迴圈用以決定重播次數或結束。

由於本程式十分短，因此幾乎可以存放在任何位置之上只要有160個BYTE空間長度。■

```
      AGE    **"
 250  VTAB 10: HTAB 7: PRINT "1) L
      OAD EXISTING SOUND TABLE"
 260  HTAB 7: PRINT "2) BEGIN NEW
      SOUND TABLE"
 270  HTAB 7: PRINT "3) INSTRUCTIO
      NS ";
 280  GET A$:X =  VAL (A$): IF X <
      1 OR X > 3 THEN 250
 290  HOME : ON X GOTO 1820,300,19
      30
 300  PRINT "ENTER NAME FOR NEW SO
      UND TABLE.": INPUT NAME$: ONERR
       GOTO 2650
 310  IF  VAL (NAME$) > 0 OR  LEN
      (NAME$) > 15 THEN  PRINT "IL
      LEGAL FILE NAME. TRY AGAIN":
       GOTO 300
 320  PRINT D$"BSAVE";NAME$;",A";S
      TADD;",L1": POKE 216,0
 330  REM  CLEAR FOR NEXT SOUND
 340  HGR : HCOLOR= 3: HPLOT 0,0 TO
      279,0 TO 279,159 TO 0,159 TO
      0,0
 350  GOSUB 1560:TNUM = 1
 360  FOR N = 0 TO 255: POKE STADD
       + SNUM * 256 - 256 + N,0: NEXT
      : REM IF ERROR, SEE LINE 186
      0
 370  HOME : VTAB 21: PRINT "RETUR
      N FOR MENU";
 380  HTAB 25: INVERSE : PRINT "SO
      UND NUMBER ";SNUM: NORMAL
 390  IF I = 1 THEN  PRINT "KEYS Y
      UI FOR UP, BNM FOR DOWN, S F
      OR DOT";: GOTO 410
 400  PRINT "PADDLE AND BUTTON TO
      SET PITCH"
 410  PRINT "SPACE BAR FOR PAUSE";
 420  HTAB 26: PRINT "ESC FOR SHIF
      T"
 430  PRINT "KEYS 0-9 AND ARROW KE
      YS TO PLACE CURSOR";
 440  REM  MAIN SOUND MAKING LOOP
 450  IF I = 1 THEN 580
 460  REM  PADDLE ENTRY
 470  IF M = 1 THEN 510
 480 N =  INT (( PDL (1) + 21.23) /
      21.24):PITCH = Q(N)
 490  HCOLOR= (N < 1): HPLOT 10,Y(
      0) TO 269,Y(0)
 500  HCOLOR= (N < 3): HPLOT 10,Y(
      1) TO 269,Y(1): GOTO 520
 510 PITCH =  PDL (1) + 1: IF PITC
      H > 255 THEN PITCH = 255
 520 CX = Z + TNUM:CY = Z + PITCH /
      2
 530  XDRAW 2 AT CX,CY: HCOLOR= 0
 540 X =  PEEK ( - 16384): IF X >
      127 THEN  POKE  - 16368,0: GOTO
      890
 550 X =  PEEK ( - 16286): IF X >
      127 THEN 620
 560  XDRAW 2 AT CX,CY: GOTO 470
 570  REM  KEYBOARD ENTRY
 580 CX = Z + TNUM:CY = Z + PITCH /
      2: XDRAW 2 AT CX,CY
 590 X =  PEEK ( - 16384): IF X >
      127 THEN  POKE  - 16368,0: GOTO
      720
 600  GOTO 590
 610  REM  PROCESS BUTTON PRESS
 620  XDRAW 2 AT CX,CY
 630 A = STADD + SNUM * 256 + TNUM
       - 256
 640  HPLOT CX, PEEK (A) / 2 + Z
 650  POKE A,PITCH: POKE A - TNUM,
      TNUM
 660 TNUM = TNUM + 1 - (TNUM > 254
      )
 670  XDRAW 1 AT CX,CY
 680  IF  PEEK ( - 16286) < 128 OR
      I = 1 THEN 450
 690  GOTO 680
 700  REM  PROCESS KEY PRESS
 710  REM  KEYBOARD ENTRY
 720  IF X < 193 THEN 890
 730  IF X = 211 THEN 620
 740  XDRAW 2 AT CX,CY
 750 X =  - 50 * (X = 217) - 10 *
      (X = 213) - 2 * (X = 201) +
      50 * (X = 194) + 10 * (X = 2
      06) + 2 * (X = 205)
 760  IF M = 2 THEN 800
 770 PITCH = PITCH + X: IF PITCH <
```

以軟體來達成的聲音合成器

```
   1 THEN PITCH = 1
780  IF PITCH > 255 THEN PITCH =
     255
790  GOTO 450
800  FOR N = 0 TO 13: IF PITCH >
     Q(N) THEN  NEXT
810 X =  SGN (X) *  INT ( ABS (X)
     ^ .4)
820 N = N + X: IF N < 0 THEN N =
     0
830  IF N > 13 THEN N = 13
840 PITCH = Q(N)
850  HCOLOR= (N < 1): HPLOT 10,Y(
     0) TO 269,Y(0)
860  HCOLOR= (N < 3): HPLOT 10,Y(
     1) TO 269,Y(1)
870  HCOLOR= 0: GOTO 450
880  REM  PADDLE AND KEYBOARD ENT
     RY
890  XDRAW 2 AT CX,CY
900 X = X - 128: IF X = 21 THEN 9
     90
910  IF X = 32 THEN 970
920  IF X > 47 AND X < 58 THEN 10
     20
930  IF X = 8 THEN 1030
940  IF X = 27 THEN  GOSUB 1610: GOSUB
     1300
950  IF X = 13 THEN 1060
960  GOTO 450
970 A = STADD + SNUM * 256 - 256 +
     TNUM
980  HPLOT CX, PEEK (A) / 2 + Z: POKE
     A,0
990  POKE STADD + SNUM * 256 - 25
     6,TNUM
1000 TNUM = TNUM + 1 - (TNUM > 25
     4)
1010  GOTO 450
1020 X = X - 49 + 10 * (X = 48):T
     NUM =  INT (X * 28.3): GOTO
     990
1030  IF TNUM < 2 THEN 450
1040 TNUM = TNUM - 2: GOTO 990
1050  REM  MAIN MENU
1060  HCOLOR= 3: HPLOT CX,4 TO CX
     ,9: HPLOT CX,142 TO CX,147
1070  HOME : VTAB 21: PRINT "  1)
     TEST SOUND";: HTAB 21: PRINT
     "5) DELETE SOUND"
1080  PRINT "  2) SAVE SOUND";: HTAB
     21: PRINT "6) CHANGE MODES"
1090  PRINT "  3) ADD TO SOUND";:
     HTAB 21: PRINT "7) QUIT"
1100  PRINT "  4) DISPLAY PREVIOU
     SLY CREATED SOUND ";: GET A$
1110 X =  VAL (A$): IF X < 1 OR X
     > 7 THEN 1060
1120  ON X GOSUB 1150,1230,1300,1
     330,1450,1480,1580
1130  GOTO 1060
1140  REM  SOUND TEST
1150  HOME : VTAB 21: HTAB 7: PRINT
     "**TEST SOUND**"
1160  INPUT "OVERALL PITCH (1-9)?
     ";P: IF P < 1 OR P > 9 THEN
     1160
1170  INPUT "NUMBER OF REPETITION
     S (1-255)? ";R: IF R < 1 OR
     R > 255 THEN 1170
1180  INPUT "LENGTH OF EACH TONE
     (1-255)? ";L: IF L < 1 OR L >
     255 THEN 1180
1190  PRINT "CALL 768,"SNUM","P",
     "R","L: CALL 768,SNUM,P,R,L
1200  PRINT "    <<PRESS RETURN FO
     R MENU>> ";: GET A$: IF  ASC
     (A$) = 13 THEN  RETURN
1210  GOTO 1150
1220  REM  SAVE SOUND
1230  HOME : VTAB 21: PRINT "VERI
     FY SAVE (Y/N)? ";: GET A$: IF
     A$ < > "Y" THEN  RETURN
1240  HOME : VTAB 21: PRINT "   *
     *SAVING SOUND TABLE**": PRINT
     "TABLE NAME - ";NAME$
1250 A = ( PEEK (STADD + NUM * 25
     6 - 256) = 0)
1260 X = 256 * (NUM - A):NUM = NU
     M + 1 - A
1270  PRINT D$;"BSAVE ";NAME$;",A
     ";STADD;",L";X
1280  POP : GOTO 1870
1290  REM  ADD TO SOUND
1300  HCOLOR= 0: HPLOT CX,4 TO CX
     ,9: HPLOT CX,142 TO CX,147
1310  POP : GOTO 370
1320  REM  DISPLAY PREVIOUS SOUND
1330  IF NUM < 2 THEN  RETURN
1340  HOME : VTAB 21: PRINT "DISP
     LAY WHICH SOUND NUMBER (1-"N
     UM;: INPUT ")? ";A$
1350 X =  VAL (A$): IF X < 1 OR X
     > NUM THEN 1340
1360  HPLOT 13,11 TO 267,11: HCOLOR=
     0: HPLOT Z,1 TO Z,158
1370 SNUM = X:TNUM =  PEEK (STADD
     + SNUM * 256 - 256)
1380  FOR N = 1 TO 255
1390 CX = Z + N:Q =  PEEK (STADD +
     SNUM * 256 - 256 + N):CY = Z
     + Q / 2
1400  HPLOT CX,1 TO CX,158: IF Q >
     0 THEN  XDRAW 1 AT CX,CY
```

```
1410  NEXT
1420  IF M = 2 THEN  GOSUB 1560
1430 CX = Z + TNUM: HCOLOR= 3: HPLOT
     CX,4 TO CX,9: HPLOT CX,142 TO
     CX,147: RETURN
1440  REM  DELETE SOUND
1450  HOME : VTAB 21: PRINT "VERI
     FY DELETE (Y/N)? ";: GET A$:
      IF A$ <  > "Y" THEN  RETURN

1460  POP : GOTO 340
1470  REM  CHANGE MODES
1480  HOME : VTAB 21: INVERSE : PRINT
     " CURRENT MODE ": PRINT M$(
     M): PRINT I$(I): NORMAL
1490  VTAB 22: HTAB 18: PRINT "1)
     CHANGE SOUND MODE"
1500  HTAB 18: PRINT "2) CHANGE E
     NTRY MODE"
1510  HTAB 12: PRINT "<<RETURN FO
     R MENU>> ";: GET A$
1520 X =  ASC (A$): IF X = 13 THEN
      RETURN
1530  IF X = 49 THEN M = 2 - (M =
     2): GOSUB 1560: GOTO 1480
1540  IF X = 50 THEN I = 2 - (I =
     2): GOTO 1480
1550  GOTO 1480
1560  HCOLOR= M - 1: FOR N = 2 *
     M - 2 TO 6: HPLOT 10,Y(N) TO
     269,Y(N): NEXT : RETURN
1570  REM  QUIT
1580  HOME : VTAB 21: PRINT "VERI
     FY QUIT (Y/N)? ";: GET A$: IF
     A$ <  > "Y" THEN  RETURN
1590  TEXT : HOME : END
1600  REM  SHIFT DISPLAY
1610  HCOLOR= 3: HPLOT CX,4 TO CX
     ,9: HPLOT CX,142 TO CX,147
1620  HOME : VTAB 21: PRINT "WITH
     THIS FEATURE YOU MAY SHIFT
     ALL OF"
1630  PRINT "THE DOTS THAT ARE TO
     THE RIGHT OF THE": PRINT "C
     URSOR."
1640  PRINT "SHIFT LEFT OR RIGHT
     (L OR R)? ";: GET A$: PRINT
     A$
1650  IF A$ = "R" THEN A$ = "RIGH
     T":X = 1: GOTO 1680
1660  IF A$ <  > "L" THEN 1610
1670 A$ = "LEFT":X = - 1
1680  PRINT "SHIFT HOW FAR "A$" (
     0-9)? ";: GET A$: PRINT A$;
     PRINT : PRINT : INVERSE : HTAB
1690  PRINT : PRINT : INVERSE : HTAB
     16: PRINT "SHIFTING";: NORMAL
1700 X = X * VAL (A$): ON SGN (
     X) + 2 GOTO 1710,1800,1730
1710  IF  - X >  = TNUM THEN X =
     1 - TNUM
1720  FOR N = TNUM TO 255: GOTO 1
     750
1730  IF X + TNUM > 255 THEN  RETURN
1740  FOR N = 255 - X TO TNUM STEP
     - 1
1750 Q = STADD + 256 * SNUM - 256
     + N: POKE Q + X, PEEK (Q): NEXT

1760  IF X > 0 THEN  FOR N = TNUM
     TO TNUM + X: POKE STADD + S
     NUM * 256 - 256 + N,0: NEXT
     : GOTO 1780
1770  FOR N = 255 + X TO 255: POKE
     STADD + SNUM * 256 - 256 + N
     ,0: NEXT
1780  POKE STADD + SNUM * 256 - 2
     56,TNUM + X:X = SNUM
1790  GOSUB 1360
1800  RETURN
1810  REM  LOAD SAVED SOUND TABLE

1820  PRINT "ENTER NAME OF EXISTI
     NG SOUND TABLE."
1830  PRINT "TYPE 'CAT' FOR A CAT
     ALOG": ONERR  GOTO 2660
1840  INPUT NAME$: IF NAME$ = "CA
     T" THEN 1900
1850  PRINT D$;"BLOAD";NAME$;",A"
     ;STADD: POKE 216,0
1860 NUM =  PEEK (43617) + 1: REM
     FOR PRODOS, CHANGE THE PEEK
     ADDRESS TO 48858
1870  HOME : VTAB 21: PRINT "FILE
      "NAME$" CURRENTLY"
1880  PRINT "CONTAINS "NUM - 1" S
     OUNDS."
1890 SNUM = NUM: GOTO 340
1900  PRINT D$;"CATALOG"
1910  GOTO 1820
1920  REM  INSTRUCTIONS
1930  HOME : HTAB 14: PRINT "INST
     RUCTIONS"
1940  VTAB 7: PRINT "THIS PROGRAM
      HAS TWO SOUND MAKING MODES.
     "
1950  PRINT "THE FIRST IS FOR MAK
     ING SOUND EFFECTS OF";
1960  PRINT "ALL VARIETIES, AND T
     HE SECOND IS FOR"
1970  PRINT "SYNTHESIZING MUSIC."

1980  PRINT : PRINT "TO USE MODE
      ONE YOU SIMPLY LAY OUT THE"
1990  PRINT "DESIRED SOUND ON THE
```

```
 HI-RES DISPLAY"
2000  PRINT "USING A PADDLE CONTR
OLLER OR JOYSTICK"
2010  PRINT "AND THE KEYBOARD, OR
 THE KEYBOARD ONLY."
2020  PRINT "THE TOP OF THE SCREE
N REPRESENTS HIGH"
2030  PRINT "PITCH, AND THE BOTTO
M REPRESENTS LOW"
2040  PRINT "PITCH."
2050  FOR N = 1 TO 255
2060 Q = (N < 127) * (125 - 60 *
 SIN (N * .05))
2070  POKE STADD + N,Q: NEXT
2080  GOSUB 2610: PRINT "AN EXAMP
LE MIGHT LOOK LIKE THIS."
2090  HGR : HCOLOR= 3: HPLOT 0,0 TO
 279,0 TO 279,159 TO 0,159 TO
 0,0
2100 X = 1:M = 1: POKE STADD,126:
 GOSUB 1360
2110  VTAB 21: PRINT "WHEN YOU WI
SH TO HEAR THE SOUND, GO TO"
2120  PRINT "THE MENU, AND USE TH
E 'TEST' OPTION.": GOSUB 261
0
2130  PRINT "YOU CONTROL THE OVER
ALL PITCH,"
2140  PRINT "THE NUMBER OF REPETI
TIONS,"
2150  PRINT "AND THE LENGTH OF EA
CH NOTE.": GOSUB 2610
2160  PRINT "SO, WITH THE SOUND P
ATTERN ABOVE, THIS,": CALL 7
68,1,1,1,30
2170  PRINT "AND THIS,": CALL 768
,1,2,10,1
2180  PRINT "AND THIS, ";: CALL 7
68,1,9,3,3
2190  PRINT "ARE POSSIBLE.": GOSUB
 2610
2200  PRINT "IF YOU ARE NOT SATIS
FIED WITH THE SOUND,";
2210  PRINT "USE THE 'ADD TO SOUN
D' OPTION.": GOSUB 2610
2220  PRINT "YOU MAY MOVE THE CUR
SOR HORIZONTALLY"
2230  PRINT "LEFT OR RIGHT WITH T
HE ARROW KEYS, AS"
2240  PRINT "WELL AS WITH THE NUM
BER KEYS (0-9).": GOSUB 2610
2250  PRINT "YOU MAY WRITE OVER A
N INCORRECT SECTION"
2260  PRINT "WITH DIFFERENT TONE
 DOTS, OR ERASE A"
2270  PRINT "SECTION BY SPACING O
VER IT.": GOSUB 2610
2280 TNUM = 30:CX = Z + 126:CY =
 60
2290  PRINT "YOU MAY ALSO SHIFT T
HE RIGHT HAND"
2300  PRINT "PORTION OF THE SCREE
N LEFT OR RIGHT."
2310  PRINT "PLACE THE CURSOR HER
E FOR INSTANCE, AND"
2320  PRINT "PRESS ESC TO TRY IT.
  <<PRESS ESC>> ";
2330  HCOLOR= 0: HPLOT CX,4 TO CX
,9: HPLOT CX,142 TO CX,147:C
X = Z + TNUM: XDRAW 2 AT CX,
CY
2340  GET A$: IF  ASC (A$) <  > 2
7 THEN 2340
2350  XDRAW 2 AT CX,CY: GOSUB 161
0
2360  HOME : VTAB 21: IF X = 0 THEN
 PRINT "PLEASE TRY AGAIN WIT
H A NON-ZERO SHIFT.": GOSUB
2610: GOTO 2320
2370  PRINT "NOTE THAT THE SOUND
READER READS ONLY UP";
2380  PRINT "TO THE 'END OF SOUND
' SLASHES.": CALL 768,1,2,3,
70
2390  GOSUB 2610: PRINT "SO PLACE
 THE CURSOR AT THE END OF TH
E"
2400  PRINT "SOUND BEFORE TESTING
.": GOSUB 2610
2410  PRINT "MODE TWO SETS UP A M
USICAL SCALE WHICH"
2420  PRINT "ALLOWS YOU TO TRANSC
RIBE DIRECTLY FROM"
2430  PRINT "SHEET MUSIC, TO A FO
RMAT LIKE THIS."
2440 Q = 1: FOR N = 0 TO 8
2450  READ X,A: DATA 7,9,5,11,3,1
4,14,1,5,9,6,8,5,9,7,7,14,60
2460  FOR TNUM = Q TO Q + A: POKE
 STADD + TNUM,Q(X)
2470  NEXT TNUM:Q = TNUM
2480  NEXT N
2490  POKE STADD,79:X = 1:M = 2: GOSUB
 1360
2500  GOSUB 2610: PRINT "NOTE THA
T HIGH NOTES PLAY FASTER SO
THAT";
2510  PRINT "THE NOTE LINE MUST B
E LONGER FOR EQUAL"
2520  PRINT "DURATION.": CALL 768
,1,4,1,12
2530  GOSUB 2610: CALL 768,1,1,1,
25: TEXT : VTAB 7
```

```
2540  PRINT "WHEN YOU ARE SATISFI
     ED WITH THE SOUND,"
2550  PRINT "RECORD THE CALL COMM
     AND DISPLAYED, AND"
2560  PRINT "SAVE THE SOUND.  TO
     USE THE SOUND IN AN"
2570  PRINT "APPLESOFT PROGRAM, B
     LOAD THE SOUND FILE"
2580  PRINT "AND THE SOUND READER
      PROGRAM, AND USE"
2590  PRINT "THE CALL COMMAND TO
     EXECUTE THE SOUND."
2600  GOSUB 2610: RESTORE : GOTO
      160
2610  VTAB 24: HTAB 9: PRINT "<<R
     ETURN TO CONTINUE>> ";
2620  GET A$: IF  ASC (A$) <  > 1
     3 THEN 2610
2630  HOME : VTAB 21: RETURN
2640  REM  DISK ERROR TRAP
2650 EL = 1: GOTO 2670: REM  ENTR
     Y POINT FOR DISK WRITE
2660 EL = 2: REM  ENTRY POINT FOR
      DISK READ
2670 ER =  PEEK (222)
2680  IF ER = 4 THEN  PRINT "DISK
      WRITE PROTECTED": GOTO 2760
2690  IF ER = 6 THEN  PRINT "FILE
      NOT FOUND": GOTO 2760
2700  IF ER = 8 THEN  PRINT "DISK
      I/O ERROR": GOTO 2760
2710  IF ER = 9 THEN  PRINT "DISK
      FULL": GOTO 2760
2720  IF ER = 10 THEN  PRINT "FIL
     E LOCKED": GOTO 2760
2730  IF ER = 11 THEN  PRINT "ILL
     EGAL FILE NAME": GOTO 2760
2740  IF ER = 13 THEN  PRINT "FIL
     E TYPE MISMATCH": GOTO 2760
2750  PRINT "ERROR #";ER"IN LINE
     "; PEEK (218) +  PEEK (219) *
      256
2760  ON EL GOTO 300,1820
```

```
:ASM
                ********************************
                *          SOUND.READER        *
                *                              *
                *      COPYRIGHT (C) 1985      *
                *      BY COMPUTING AGE        *
                *    ASSEMBLER: BIG  MAC       *
                ********************************
                         ORG   $0300
                SNUM     EQU   $06
                OPITCH   EQU   $07
                REP      EQU   $08
                TLEN     EQU   $09
                TNUM     EQU   $1C
                TNUM2    EQU   $1B
                TLEN2    EQU   $1A
                STSOUND  EQU   $FF00
                COMBYTE  EQU   $E74C
                SPKR     EQU   $C030
                *
                * STORE SOUND NUMBER
0300: 20 4C E7           JSR   COMBYTE
0303: 86 06              STX   SNUM
                * STORE OVERALL PITCH
0305: 20 4C E7           JSR   COMBYTE
0308: 86 07              STX   OPITCH
                * STORE NUMBER OF REPETITIONS
030A: 20 4C E7           JSR   COMBYTE
030D: 86 08              STX   REP
                * STORE LENGTH OF EACH TONE
030F: 20 4C E7           JSR   COMBYTE
0312: 86 09              STX   TLEN
                * CALCULATE HIGH BYTE OF THE
                * ADDRESS OF THE BEGINNING
                * OF THE SOUND
0314: A0 64              LDY   #$64
0316: 88                 DEY
0317: C8        INCADD   INY
0318: C6 06              DEC   SNUM
031A: D0 FB              BNE   INCADD
                * CORRECT STSOUND
031C: 8C 24 03           STY   $0324
031F: 8C 3F 03           STY   $033F
                * GET SLEN FROM SOUNDFILE
0322: AE 00 FF           LDX   STSOUND
0325: 86 1C              STX   TNUM
                * CALCULATE AND CORRECT 'BRANCH'
                * BASED ON THE VALUE OF THE
                * OVERALL PITCH
0327: A0 00              LDY   #$00
0329: 88        CHBR     DEY
032A: 88                 DEY
032B: 88                 DEY
032C: C6 07              DEC   OPITCH
032E: D0 F9              BNE   CHBR
0330: 8C 61 03           STY   $0361
                *
                * SOUND PRODUCING LOOP
                *
0333: A0 01     REPEAT   LDY   #$01
0335: A6 1C              LDX   TNUM
0337: 86 1B              STX   TNUM2
0339: A5 09     NEXTPN   LDA   TLEN
033B: 85 1A              STA   TLEN2
                * LOAD X WITH A PITCH NUMBER
                * FROM SOUNDFILE
033D: BE 00 FF  SAMEPN   LDX   STSOUND,Y
0340: E8                 INX
0341: CA                 DEX
                * IF PITCH NUMBER=0 SKIP CLICK
0342: F0 1B              BEQ   DECPIT
                * CLICK SPEAKER
0344: AD 30 C0           LDA   SPKR
                * PAUSE BETWEEN CLICKS
                * (LONG PAUSE = LOW PITCH)
0347: 4C 4A 03           JMP   J1
034A: 4C 4D 03  J1       JMP   J2
034D: 4C 50 03  J2       JMP   J3
0350: 4C 53 03  J3       JMP   J4
0353: 4C 56 03  J4       JMP   J5
0356: 4C 59 03  J5       JMP   J6
0359: 4C 5C 03  J6       JMP   J7
035C: 4C 5F 03  J7       JMP   DECPIT
035F: CA        DECPIT   DEX
                * IF X<>0, CONTINUE PAUSE
0360: D0 FE     BRANCH   BNE   BRANCH
0362: C6 1A              DEC   TLEN2
                * IF TLEN2<>0, SAME PITCH NUMBER
0364: D0 D7              BNE   SAMEPN
0366: C8                 INY
0367: C6 1B              DEC   TNUM2
                * IF TNUM2<>0, NEXT PITCH NUMBER
0369: D0 CE              BNE   NEXTPN
036B: C6 08              DEC   REP
                * IF REP<>0, REPEAT SOUND
036D: D0 C4              BNE   REPEAT
036F: 60                 RTS

--End assembly--

112 bytes

Errors: 0
```

機械語言入門　吳博士

機械語言對高解像度圖形的使用(上)

在以往我們曾經討論過低解像度圖形，而今次我要討論高解像圖形，其中包括①高解像度圖形如何貯存在記憶體裏，②色彩的構成，③在高解像度圖形上繪劃點及線。若果對以上各項充份掌握的話，利用這些技巧可以產生不同的效果出來。

高解像度圖形的記憶區域及選擇顯示模式

大家都知道高解像圖形有兩頁，它們共佔長達16K的記憶體，每一頁佔用8K，第一頁由＄2000開始至＄3FFF止，第二頁由＄4000至5FFF止。若果我們要選擇這兩頁圖形或其它的模式，可以利用圖一所示的軟體開關來達成，列表1的程式是用來選擇低解像度圖形或高解像度圖形的顯示，其中我們只需按下“B”鍵便可以將這兩種模式交替顯示。

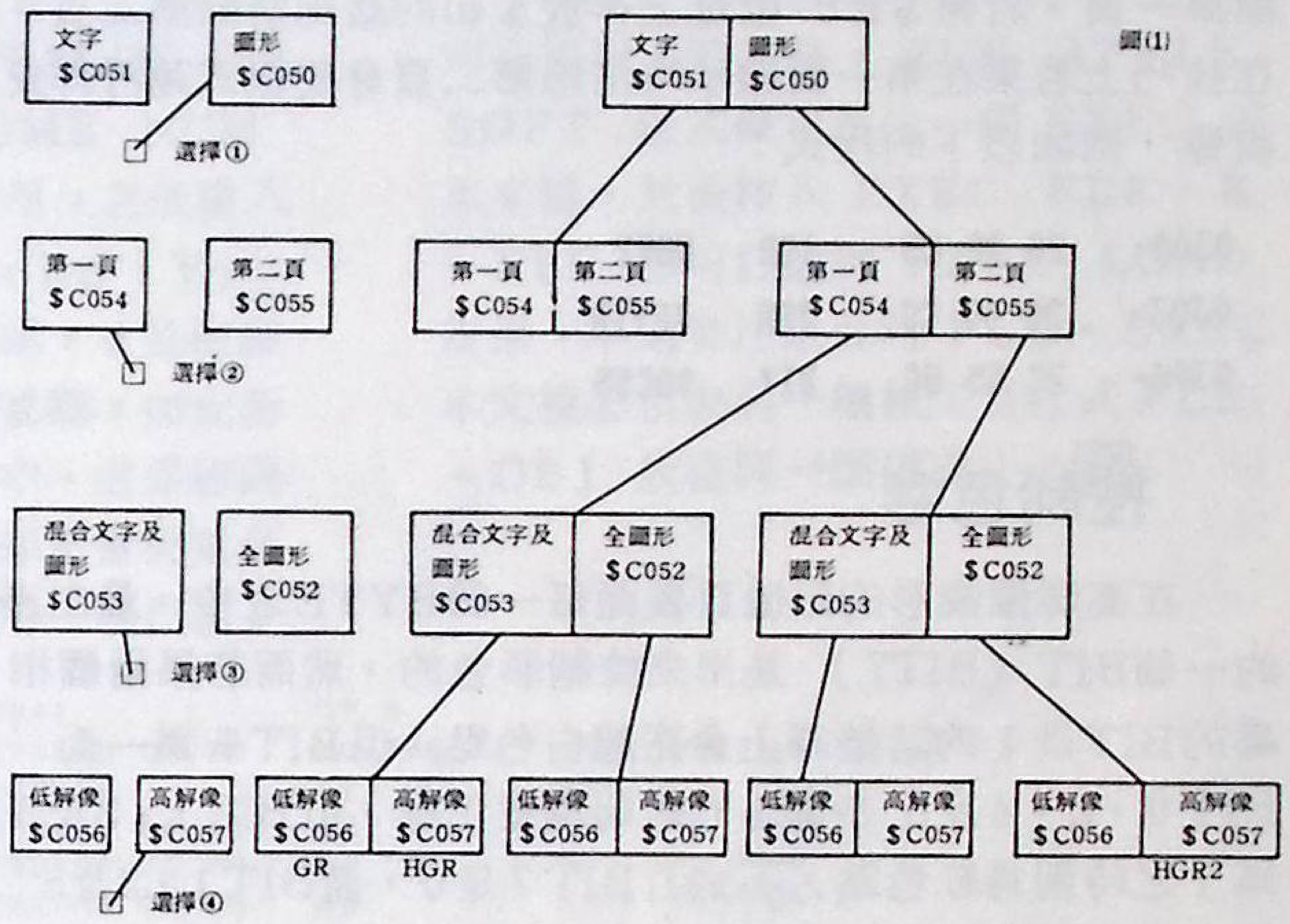
一如圖一所示，軟體開關可以選擇不同的顯示模式，以高解像圖形來說就有著四種不同的模式，它們分別是第一頁／第二頁的選擇及混合圖形文字／全圖形顯示模式。

每一頁高解像度圖形螢幕由280個水平點及192個垂直點構成，當混合圖形文字模式時垂直點數目為160點。

列表(1)

```
30D : 2C 50  C0     BIT $C050       : 選擇圖形
                 REPEAT
310 : 2C 57  C0     BIT $C057       : 選擇高解像度
313 : 20 00  C0     JSR KEYIN       : 等待按鍵
316 : C9 C2         CMP # "B"       : 若果按下B的話
318 : D0 0A         BNE  DONE       : 則
31A : 2C 56  C0     BIT $C056       : 選擇低解像度
31D : 20 00  C0     JSK KEYIN       : 等待按鍵
320 : C9  C2        CMP # "B"       : 若果按下B
322 : F0 EC         BEQ REPEAT      : 則到$310再來
324 : 60            RTS             : 結束
```

APPLE 內置的副程式



圖(1)

APPLE 的唯讀記憶體（ROM)內其實已有著一組處理高解像度圖形的副程式，我們可以直接利用機械語言來使用它們，這些副程式佔用了第零頁記憶體內的一些位址，它們是：

＄E0 —貯存一點在螢幕上的，水平座標之低位元組。

＄E1 —貯存一點在螢幕上的，水平座標之高位元組。

＄E2 —貯存一點在螢幕上的垂直座標。

＄E4 —貯存顏色代碼。

＄E6 —貯存所要選用的高解像度頁數（＄20代表第一頁，＄40代表第二頁）。

這些程式共有七個，它們可歸納於表2之中：

從表(2)我們已經看到APPLE 內置的副程式及功用，不過它們具體的使用細則却不大了然，至此我們回想一下，究竟高解像圖形是如何構成的。

任何色彩的高解像圖形其實都是由不同顏色的點構成，因此若果我們懂得如何控制每點的色彩，及懂得在螢幕上繪出它們，理論上任何圖形都可繪出，以下是這些副程式的應用例子。

清除高解像度螢幕

名稱	位址	累加器	X暫存器	Y暫存器	附　註
HGR.	＄F3E2.				表(2)
HGR2.	＄F3D8				
BKGND	＄F3F4	色彩位元			
HCOLOR	＄F6F0		色彩由0至7		
HPLOT	＄F457	垂直位置	水平低位元	水平位置高位元	此程式會呼叫HPOSN
HLINE	＄F53A	水平位置低位元	水平位置高位元	垂直位置	由最後一次位置開始繪出直線
HPOSN	＄F411	垂直位置	水平位置低位元	水平位置高位元	此程式必須在執行其他程式前叫用

機械語言對高解像度圖形的使用(上)

將高解像螢幕清除其實是將黑色塡滿螢幕（有關色彩問題控制稍後會討論），亦即將＄00貯在＄2000至＄3FFF 或＄4000 至＄5FFF 之上，內置副程式HCLR（在＄F3F2）是用作這個功用。只要將HPAG（＄E6）上放入＄20即會清除第一頁，若果＄E6 位址上存放＄40的話則清除第二頁。在技巧上若果在第一頁顯示時清除第二頁會使程式運行得更流暢，例如以下的程式：

```
0300-   20 DE F3   JSR   HGR2
0303-   20 00 C0   JSR   KEYIN
0306-   2C 55 0C   BIT   $0C55
```

控制色彩

在高解像圖形的記憶頁裏面每一個BYTE之中，最左邊的一個BIT（BIT7） 是用來控制彩色的，然而若果兩個相鄰的BIT是1的話螢幕上會亮起白色點，但BIT７爲一而BIT０，２，４及６ 亦是１的話則爲藍色點，但BIT１，３及5爲１之時則爲紅色點，不過若BIT７是0，而BIT１，３及5爲１時則亮起紫色，而BIT１，３，５，７ 同時爲１時則亮起綠色，以上這些結果組成了高解像圖形的六種顏色。

彩色BIT

	0	1
偶數BIT	紫色	藍色
單數BIT	綠色	紅色
相鄰各點爲１時	白色	

各位若果要看看這些不同的色彩的話，只要將列表(3)的程式＄304之內的＄37 改爲適當的數目，其中＄55會亮起４個紫色點，$AA亮起３個爲紅色點，而＄7F及＄FF 則產生７個白色點。

內置程式之中BKGND（＄F3F6 之上）可以將高解像記憶區域，塡滿不同的底色，只需將HPAG（＄E6)存入頁數（＄20爲第一頁，＄40爲第二頁），HCOLOR１（$1C）存入彩色（＄55紫色，＄2A綠色，＄D5藍色，＄AA紅色，＄7F或＄FF爲白色，＄00爲黑色），列表(4)中的程式用作將顏色塡滿螢幕。

列表(3)

```
0300-   20 E2 F3   JSR   HGR     ；設定及清除第一頁
0303-   A9 37      LDA   #$37    ；存入00110111
0305-   8D 00 20   STA   $2000   ；到＄2000上
0308-   20 00 C0   JSR   KEYIN
030B-   60         RTS
```

列表(4)

```
0300-   20 E2 F3   JSR   HGR     ；設定及清除第一頁
0303-   A9 55      LDA   #$55    ；存入01010101
0305-   85 1C      STA   HCOLOR  ；到$1C之上
0307-   20 F6 F3   JSR   BKGND   ；塡滿螢幕
030A-   20 00 C0   JSR   KEYIN   ；等待按鍵
030D-   60         RTS           ；結束
```

繪出點的方法

高解像圖形的螢幕上分爲280 ×192 點或（280 ×160 點混合文字模式），其中X座標由＄00至＄117（0—279）Y座標由＄00至＄BF（0至191），在你繪出點之前必須在HCOLOR（＄E4）之上存入彩色數目，這些數目是我們以上所討論過的，然後將Y座標存入累加器上，X 座標低位元存入X 暫存器之上，而X座標的高位元存入Y暫存器之中，之後可以呼叫內置程式HPLOT（＄F457），若果你要繪出一個綠色點或紅色點在偶數行裏，或紫色／藍色點到奇數行裏的話，只要更改X ，Y 暫存器的值，列表(5)的程式會在螢幕中央繪出白色點。

列表(5)

```
0300-   20 E2 F3   JSR   HGR     ；設定並清除第一頁
0303-   A9 FF      LDA   #$FF    ；白色
0305-   85 E4      STA   HCOLOR
0307-   A9 50      LDA   #$50    ；Y＝80
0309-   A2 8C      LDX   #$8C    ；X ＝140
030B-   A0 00      LDY   #$00
030D-   20 57 F4   JSR   HPLOT   ；繪出此點
0310-   60         RTS             結束
```

繪出直線的使用

內置程式之中HLIN（在＄F53A上）會繪出直線，由最是一次繪出的點開始至X ，Y 爲止，在呼叫HLIN 之前，將Y 存入Y暫存器內，X 座標的低位元存入累加器，而高位元存入X暫存器上。爲了應用HLIN ，我們通常都使用HPLO—T 先繪出一點再用HLIN 繪出直線至新座標去，列表(6)的程式加在HPLOT 程式之後時會先在X＝140，Y＝80 上繪出一個白點，再由它繪出兩條線到（32，144）及（112，16）上。

列表(6)

```
0313-   A9 20      LDA   #$20    ；X＝32
0315-   A2 00      LDX   #$00
0317-   A0 90      LDY   #$90    ；Y＝144
0319-   20 3A F5   JSR   HLIN    ；繪出直線
031C-   20 00 C0   JSR   KEYIN   ；等候按鍵
031F-   A9 70      LDA   #$70    ；X＝112
0321-   A2 00      LDX   #$00
0323-   A0 10      LDY   #$10    ；Y＝16
0325-   20 3A F5   JSR   HLIN    ；繪出直線
0328-   20 00 C0   JSR   KEYIN   ；等候按鍵
032B-   60         RTS           ；結束
```

由於篇幅所限，今次我們就到此爲止，下期將會研究高解像圖形表的應用，使各位可以利用機械語言來控制自己創作的任何圖形，而今次所討論的應用將以同期的〔高解像圖魔術師〕一文中的程式作爲實踐，我將會詳細爲各位分析其程式的結構，請參看〔高解像圖魔術師」一文。■

DOS 3.3 專欄

甘貴祥

DOS的快速存取

本程式是可以將標準 DOS 的 LOAD 或 BLOAD 功能提高，使速度快2－5倍，但却完全解決軟件的相融問題。

本程式包含兩個部份，一個是可以永久放在 DOS 3.3 內的機械語言副程式，另一個是 APPLE SOFT程式，用來產生一個本文檔作 EXEC 使用，它可將副程式放入DOS 之內做一些小的修改。

要製造永久的 FAST LOAD 磁碟，首先取一片已 INIT 過的磁碟（35 或 40 TRACK 均可），而上面的資料是不要的，用 CATALOG 命令檢查過並抄下它的 VOLUME NUMBER 以便 INIT之用，之後鍵入 INIT HELLO，Ss，Dd，Vv

其中S是 SLOT 號碼，d是磁碟號碼，V是 VOLUME 號碼，如此新的 DOS 便被寫入磁碟之中，這張磁碟會被重定爲初設狀態，但却不會被重新FORMAT，因此你需要用一張已經FORMAT 過的磁碟。

假如你不希望這種處理方法的話，可以用以下的方法，RUN APPLE SOFT 程式使它產生一個 EXEC. 的本文檔，然後按入 EXEC FLE・EXEC 即可以達到 FAST LOAD 效果，不過要注意此時 FLE・EXEC 本文檔必須與另一機械語言程式 FLE・OBJ 放在同一面磁碟上。■

```
SOURCE FILE: FLE
0000:          *
0000:          * FAST LOAD ENHANCEMENT *
0000:          *
0000:          * DOS 3.3 LOAD/BLOAD EXT_ENSIONS *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
0000:          *
B5BB:          FMPL     EQU  $B5BB
B5D1:          FMWA     EQU  $B5D1
B39B:          STKSAVE  EQU  $B39B
B5E4:          FILPOSN  EQU  $B5E4
B7E8:          IOBORG   EQU  $B7E8
B7F0:          IOBBUFA  EQU  IOBORG+8
0042:          BUFADR   EQU  $42
0048:          STBUF    EQU  $48
0000:          *
0000:          *
0000:          *
B052:          RWTSDRVR EQU $B052
AF10:          SELDABF  EQU  $AF10
AF0C:          SELTSBF  EQU  $AF0C
AF5E:          RDTSLIST EQU $AF5E
B385:          SETERROR EQU $B385
0000:          *
BE0D:          FORMDSK  EQU  $BE0D
A40A:          RWR.2    EQU  $A40A
0000:          *
----- NEXT OBJECT FILE NAME IS FLE.OBJ0
BEAF:                   ORG  $BEAF
BEAF:          *
BEAF:          * WORKING STORAGE FOR EAST LOADER
BEAF:          *
B5E6:          FL.OFFST EQU FILPOSN+2
B5C3:          FL.FWA   EQU  FMPL+8
B5C1:          FL.LDLN  EQU  FMPL+6
BEAF:          *
BEAF:          * LOAD/BLOAD ENTRY - CALLED AFTER THE
BEAF:          * FIRST 2 OR 4 BYTES HAVE BEEN READ
BEAF:          *
BEAF:D8        FLE      CLD
BEB0:BA                 TSX
BEB1:8E 9B B3           STX  STKSAVE
BEB4:          *
BEB4:AD C3 B5  FLE1     LDA  FL.FWA
BEB7:38                 SEC
BEB8:ED E6 B5           SBC  FL.OFFST
BEBB:8D C3 B5           STA  FL.FWA
BEBE:AD C4 B5           LDA  FL.FWA+1
BEC1:E9 00              SBC  #0
BEC3:8D C4 B5           STA  FL.FWA+1
BEC6:          *
BEC6:A9 0E              LDA  #$E
BEC8:8D 65 BF           STA  TSPTR
BECB:20 10 AF           JSR  SELDABF
BECE:          *
BECE:AC E6 B5           LDY  FL.OFFST
BED1:AD C1 B5           LDA  FL.LDLN
BED4:18                 CLC
BED5:6D E6 B5           ADC  FL.OFFST
BED8:8D C1 B5           STA  FL.LDLN
BEDB:A9 00              LDA  #0
BEDD:6D C2 B5           ADC  FL.LDLN+1
BEE0:8D C2 B5           STA  FL.LDLN+1
BEE3:          *
BEE3:F0 5F              BEQ  FLE9
BEE5:          *
BEE5:          *COPY THE REST OF THE FIRST SECTOR
BEE5:          * FROM THE I/O BUFFER TO MEMORY
BEE5:          *
BEE5:20 5A BF           JSR  SETSTBF
BEE8:B1 42     FLE2     LDA  (BUFADR),Y FETCH BYTTE
BEEA:91 48              STA  (STBUF),Y
BEEC:C8                 INY
BEED:D0 F9              BNE  FLE2
BEEF:          *
BEEF:          *WE HAVE MOVED THE REST OF THE FIRST
BEEF:          *SECTOR TO THE PROPER PLACEIN MEMORY
BEEF:          *NOW WE READ WHOLE SECTORS UNTIL LESS
BEEF:          *THAN A SECTTOR OF DATA IS LEFT.
BEEF:          *
BEEF:EE C4 B5  FLE3     INC  FL.FWA+1
BEF2:20 0C AF           JSR  SELTSBF
BEF5:AC 65 BF           LDY  TSPTR
BEF8:D0 08              BNE  FLE6
BEFA:38                 SEC
BEFB:20 5E AF              JSR       RDTSLIST
BEFE:B0 55              BCS  FLEERR
BF00:A0 0C              LDY  #$C
BF02:          *
BF02:B1 42     FLE6     LDA  (BUFADR),Y
BF04:AA                 TAX
BF05:C8                 INY
BF06:B1 42              LDA  (BUFADR),Y
BF08:C8                 INY
BF09:8C 65 BF           STY  TSPTR
BF0C:A8                 TAY
BF0D:CE C2 B5           DEC  FL.LDLN+1
BF10:F0 15              BEQ  FLE8
BF12:          *
BF12:AD C3 B5           LDA  FL.FWA
BF15:8D F0 B7           STA  IOBBUFA
BF18:AD C4 B5           LDA  FL.FWA+1
```

```
BF1B:8D F1 B7   119          STA  IOBBUFA+1
BF1E:A9 01      120          LDA  #1
BF20:20 52 B0   121          JSR  RWTSDRVR
BF23:90 CA      122          BCC  FLE3
BF25:B0 2E      123          BCS  FLEERR
BF27:           124 *
BF27:           125 *WE ARE HERE TO READ THE LAST SECTOR
BF27:           126 *INTO THE DOS I/O BUFFER AND THEN
BF27:           127 *COPY THE REMNANT TO THE END OF THE
BF27:           128 *THE CODE IMAGE IN MEMORY.
BF27:           129 *
BF27:AD C1 B5   130 FLE8     LDA  FL.LDLN
BF2A:F0 27      131          BEQ  FLE11
BF2C:8A         132          TXA
BF2D:48         133          PHA
BF2E:20 10 AF   134          JSR  SELDABF
BF31:A5 42      135          LDA  BUFADR
BF33:8D F0 B7   136          STA  IOBBUFA
BF36:A5 43      137          LDA  BUFADR+1
BF38:8D F1 B7   138          STA  IOBBUFA+1
BF3B:           139 *
BF3B:68         140          PLA
BF3C:AA         141          TAX
BF3D:A9 01      142          LDA  #1
BF3F:20 52 B0   143          JSR  RWTSDRVR
BF42:A0 00      144          LDY  #0
BF44:           145 *
BF44:           146 *WE ARE HERE WITH THE LAST PARTT OF
BF44:           147 *THE DATA IN THE DOS I/O BUFFER
BF44:           148 *
BF44:20 5A BF   149 FLE9     JSR  SETSTBF
BF47:CC C1 B5   150 FLE10    CPY  FL.LDLN
BF4A:B0 07      151          BCS  FLE11
BF4C:B1 42      152          LDA  (BUFADR),Y FETCH BYTTE
BF4E:91 48      153          STA  (STBUF),Y STORE BYTE
BF50:C8         154          INY
BF51:D0 F4      155          BNE  FLE10
BF53:           156 *
BF53:           157 * ALL DONE WITH THE LOAD
BF53:           158 *
BF53:18         159 FLE11    CLC
BF54:60         160          RTS
BF55:           161 *
BF55:A9 08      162 FLEERR   LDA  #8
BF57:4C 85 B3   163          JMP  SETERROR
BF5A:           164 *
BF5A:AD C3 B5   165 SETSTBF  LDA  FL.FWA
BF5D:85 48      166          STA  STBUF
BF5F:AD C4 B5   167          LDA  FL.FWA+1
BF62:85 49      168          STA  STBUF+1
BF64:60         169          RTS
BF65:           170 *
BF65:00         171 TSPTR    DFB  0          T/S LIST POINTER
00B7:           172 LENFLE   EQU  *-FLE

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

```
10  REM  ********************
40  REM  * FAST LOAD ENHANCE*
50  REM  *EXEC INSTALL CREAT*
70  REM  * COPYRIGHT (C)1984*
90  REM  ********************
108 D$ =  CHR$ (4)
110  REM  * OPEN THE TEXT FILE
120  REM  *                  *
130  PRINT D$;"OPEN FLE.EXEC "
140  PRINT D$;"WRITE FLE.EXEC "
150  REM  * LOAD FLE SUBROUTINE
160  PRINT "BLOAD FLE.OBJ ,A$BEAF
     "
170  REM  * CHANGE LOAD BLOAD TO
     CALL FLE
180  PRINT "CALL-151"
190  PRINT "A40B: AF BE "
200  REM  * DISABLE RWTS FORMAT
210  PRINT "BEOD :18 60 "
220  REM  * RETURN TO BASIC  *
230  PRINT "3DOG"
240  PRINT D$;"CLOSE FLE.EXEC "
250  END
```

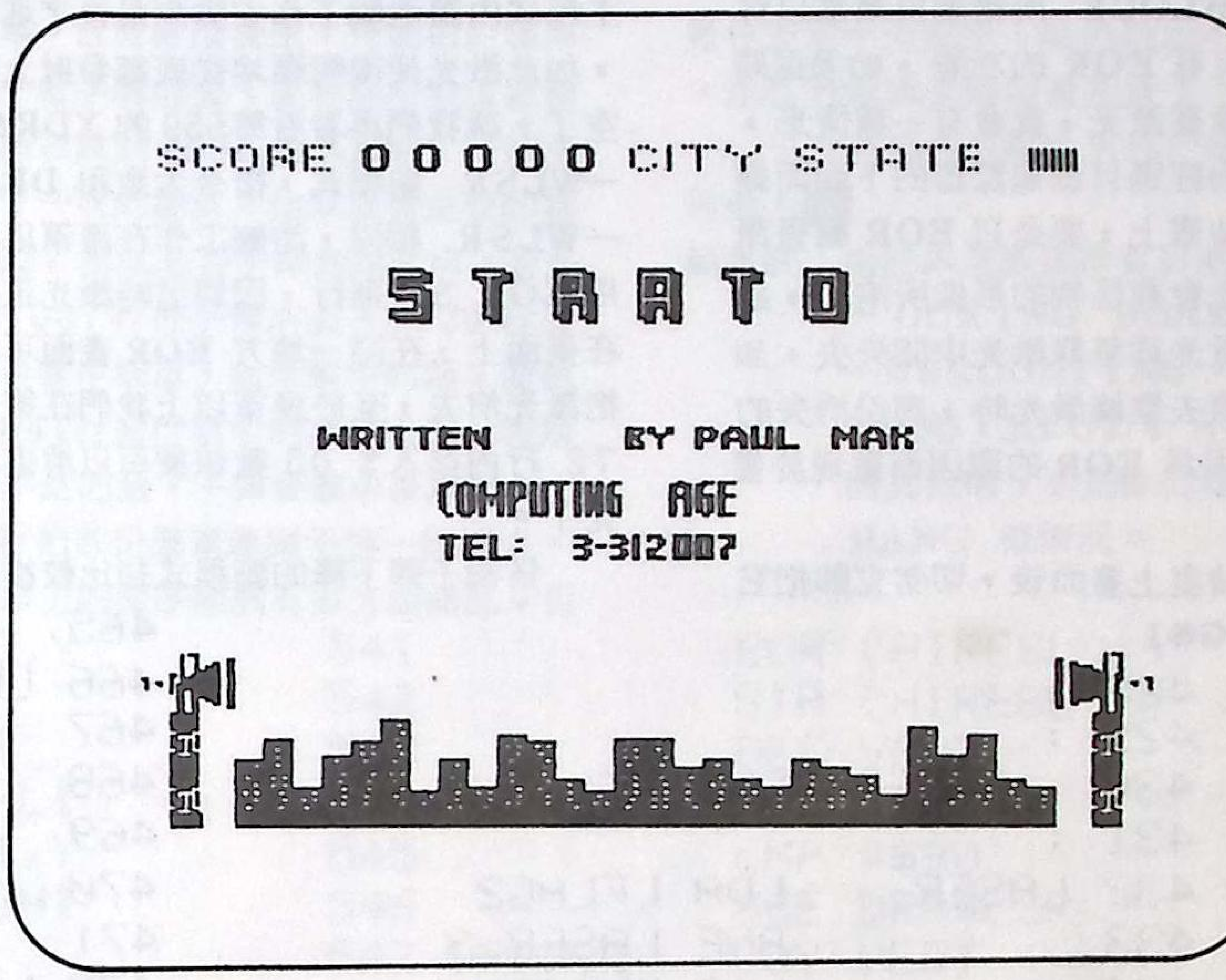
16 - A7

GRAPHICS MAGAICIAN編寫的遊戲：
STRATO 城市保衛戰

麥法基

這一期筆者把一個 ATARI 的電腦遊戲改編在 APPLE 上進行。由於配上 MOCKINGBOARD的關係，音响方面可說是多姿多釆。假如讀者到電腦時代購買磁碟，當磁碟 BOOT 起時，程式會自動找尋該 CARD，如果沒 MOCKINGBOARD 遊戲會改由 APPLE SPEAKER 發聲。

遊戲很簡單，玩者控制螢幕上的一個十字形目標追踪器，按 BUTTON 0 時會有激光從下面的城市射出，對象當然是攻擊城市的怪物了。遊戲開始時城市之上會有護罩阻止怪物的子牌。當護罩被子彈擊中而消失後，城市便會受到破壞，當城市損壞程度達到危機程度時整個遊戲會完結。不過如果把七種不同的怪物擊退後，城市和護罩都會重新再為你建造。



遊戲採用 BLOCK 繪圖程式作爲動畫部份，但基於 APPLE 高解像畫面的特殊結構，假如全部採用 BLOCK 繪圖法，勢必影响畫面上某些背景。最主要是護罩的保留與消失，因爲當激光從畫面城市發射無須消去護罩，但怪物的子彈却會消去部份護罩，所以激光的繪畫分爲兩部份，而很明顯地怪物的子彈亦須要兩種繪圖方法進行。

任何畫面上物體移動都是基於一個原理，畫圖——消圖——在新的地方畫圖。所以無論用甚麽方法畫圖與消圖，都可以達到物體移動的效果，採用 BL—OCK 的繪圖法大大地增加程式運行速度，但不幸地却缺少了 XDRAW 的碰撞檢知方法，而這種碰撞檢知是護罩和城市被子彈擊中所十分需要的。筆者於是把兩種繪圖方法都同時加以應用，在護罩以上的畫面均以 BLOCK 的方法繪圖，而護罩以下則以 XDRAW 方法繪圖，現在看看如何繪畫激光部份：

參看 LISTING #1 的 STR—AT 0 部份程式。第449行的 VER—T是激光開始垂直座標（Y—AXIS），因爲激光永遠從城市底部向天空發射，所以設定爲$ B 7 。而 450 行的 MKVERT和455行的 MKXBYTE 分別爲目標追踪器的垂直座標及水平位元，亦是經常移動部份。每次按 BUT—TON 時程式便須要把一條直綫從$ B7 一直繪畫到 MKVERT 和 XBYTE 的位置上，達成激光發射的效果，當然亦要把這激光消去。如果以 BASIC 或採用 ROM 内的 HLINE（$F53A）ROUTINE 當然是輕而易擧的事，問題是有護罩和城市的存在，利用該等 ROUTINE 會洗去部份護罩和城市，而且遊戲用去大部份零頁區，絕無可能再加上 ROM的 ROUTINE。

到目前爲止我們知道目標追踪器的正確位置，不過仍然要作出一些位置上的修改，追踪器共佔三個位元，而激光只要繪畫至追踪器底部便可，所以把M—KVERT加上$ C 暫存於 MKVERT 1 而MKXBYTE 加上$ 1 暫存於MK—XBYTE 1 內以供繪畫激光時終止位置所在。

再看看537行的 DRAWLSR 副程式，首先把Y— REGISTER 存入 VERT 即激光起始最底位置垂直數值，再呼叫495行的 GETADR 的找尋水平位元副程式。我們雖然已知目標追踪器的水平位元，但要憑它從 GETAD—R 裡找出和底位置對應的水平位元組才能成爲激光起始位置水平數值。至於 TABL 0 和 TABH 1 是高解像度第一頁的畫面位置高低位元組表，而H—IRESL 和 HIRESH 則爲位置的指位器。由第 539 行起便開始繪直綫工作了，直綫是一條幼白綫，所以只佔用一個位元組，設定Y— REGISTER 爲零（因爲我們利用 HIRESL的指位器，設定Y— REGISTER 爲零表示只取一個位元組，這和上述第 537行的 Y—REGISTER 值有所不同，該值只表示垂直數值，而當呼叫 GETADR 後該值無須再保留，它是由 VERT 的數值所決定。不要忘記 APPLE 只有 X和Y兩個暫存器供索引之用，故Y—REGISTER 很多時都會由數個副程式用到。）第540 行的$ 60 相等於二進制的 01100000 ，亦即等於在畫面上繪上一個白點，不過這時繪畫在護罩之下，所以必須首先 EOR（HIRESL），Y以供保留背景，然後再從542行的 STACHIRESL），Y繪上螢幕。

第 543行把 VERT値減一，把垂直位置向上移一點，這時需要檢知該值是否已經脫離護罩範圍，假如不少於$ 90的話，則至537行繼續繪畫白綫，如果少過$ 90 便至547行，這時無須

城市保衛戰

再保留背景，所以550和551行之間除去了 EOR 的指令。爲甚麽如此麻煩而不完全採用 EOR 的方法把整條激光繪畫出來呢？最主要的原因是護罩以上的怪物是由 BLOCK 所繪畫出來加以移動，它們沒有 EOR 的功能，如果這時用 EOR 繪畫激光，就會有一種情形，假設當怪物經過目標追踪器的下面而剛巧激光同時畫上，要是以 EOR 繪畫激光，則激光會被怪物的形像所消去，造成有部份激光從整條激光中間失去，加上我們要消去整條激光時，部份消失的激光却會因爲 EOR 的原因而重現於畫面上。

當激光繪畫上畫面後，切勿立即把它消去，因爲機械語言運行太快，立即把它消去會使激光不容易被察覺，所以第485行的 XLASER 把消激光的 RO—UTINE 獨立起來，這樣我們可以在主程式內讓消圖工作安排到最後才進行，如此激光便很明顯地從底部發射上天空了。讓我們再看看第559的 XDRA—WLSR 副程式，指令大致和 DRA—WLSR 相同，消圖工作在護罩以下用 EOR 方法進行，因爲這時激光正存在畫面上，在同一地方 EOR 畫面可以把激光消去，至於護罩以上我們在第572 行內存入$ 00 數值便可以消去激光了。

怪物子彈下降的副程式却比較複雜，一方面要照顧到遊戲難度增加時子彈數目會隨著增加，而一方面却要檢知護罩和城市的不同碰撞。參看 LISTIN—G ＃ 2 第972 行，NUM 是子彈數目的索引，數值由遊戲難度所決定，但不會超過十二枚，因爲子彈太多遊戲變成十分困難，根本沒有趣味可言。972至988行十分簡單，把所發子彈首先繪畫在畫面上，留意1039 行的 XAB 4 是把子彈圖形呼叫 GRAPHICS MA—GICIAN的BLOCK繪圖程式進行畫圖工作。當子彈被繪畫在畫面而存在後，第989行便是移動部份了，第997行的 SPEED 是子彈速度，亦是由難度所決定，子彈移動時我們要做幾種測

LISTING#1

```
428 ;
429 ;
430 ;  LASER ROUTINE
431 ;
432 LASER      LDA LFLAG2
433            BNE LASER.3
434            LDA BUTTON
435            BMI FIRE
436            LDA #$0
437            STA SHOT
438 LASER.1    JMP NOSHOT.1
439 FIRE       LDA SHOT
440            CMP #$1
441            BGE LASER.1
442            LDA ACFLAG
443            BNE LASER.1
444            LDA #$1
445            STA SHOT
446            STA LFLAG
447            STA LFLAG2
448            LDA #$B7
449            STA VERT
450            LDA MKVERT
451            STA MKVTEMP
452            CLC
453            ADC #$C
454            STA MKVERT1
455            LDA MKXBYTE
456            STA MKXTEMP
457            CLC
458            ADC #$1
459            STA MKXBYTE1
460 LASER.2    JSR DRAWLSR
461            LDA #$28
462            STA LPITCH
463            JSR ACOLLIS
464            JSR BCOL
465            JMP LASER.4
466 LASER.3    LDA LPITCH
467            CMP #$32
468            BGE NOSHOT
469            INC LPITCH
470            INC LPITCH
471            INC LPITCH
472 LASER.4    LDA #$2
473            STA TIME
474            LDA LPITCH
475            STA PITCH
476            JSR SOUND
477            JMP NOSHOT.1
478 NOSHOT     LDA #$0
479            STA LFLAG2
480 NOSHOT.1   RTS
481 ;
482 ;
483 ; XDRAW LASER ROUTINE
484 ;
485 XLASER     LDA LFLAG
486            BNE XLASER.1
487            RTS
488 XLASER.1   LDA #$B7
489            STA VERT
490 XLASER.2   JSR XDRAWLSR
491            LDA #$0
492            STA LFLAG
493            RTS
494 ;
495 GETADR     LDA TABLO,Y
496            CLC
497            ADC MKXBYTE1
498            STA HIRESL
499            LDA TABHI,Y
500            STA HIRESH
501            RTS
502
```

城市保衛戰

抱歉由於列表太長，這裏只刊登本遊戲的玩法，讀者可在本期程式磁碟中輕易找到遊戲的FILE 。

試，第998行測試是否在護罩範圍，若否則至1009行繼續進行 BLOCK 的移動，若是則至1016行進行XDRAW的移動（第1002行測試＄D0是在所有怪物被消滅後換版時子彈變成流星雨部份，暫時可加以不理）。讓我們觀察1016行，這是以 YDRAW 來移動子彈，DRAW是一條GRAPHICS MAGICIAN 應用的 XDRAW 繪畫程式，筆者把它節錄在第1265行，程式只要把子彈位置的水平位元組存入＄0，水平位元存入＄1，垂直數值存入＄2，而圖形號碼存入＄3，呼叫DR—AW便可把圖形繪畫上畫面。（注意這是 EOR 的繪圖）1018 行的DRA—W當然是首先把子彈在舊地方消去而1022行的 DRAW把它在新地方畫圖，這時 1023行的 COLLSN便是碰撞檢知的旗號發揮作用了當旗號爲零時十分簡單，沒有碰撞發生，可能由於護罩被上一粒子彈消去，或者部份城市被擊去，所以跳到1036的 XAB 2 以待主程式下一次的測試。如果旗號是壹的話，1027 測試是否子彈到達底部，若是則至1037行把子彈消去。1029 行測試是否擊中城市，如果擊中的話，便跳到 1041行進行擊中的其他工作。若兩者皆不是的話，子彈便擊中護罩而在1031行把部份護罩連同子彈一同消去。

以上的程序雖然有多方面測試，但由於每種情形均不會同時發生，實際上不會把遊戲速度減慢，亦由此可見機械語言的速度實在驚人。

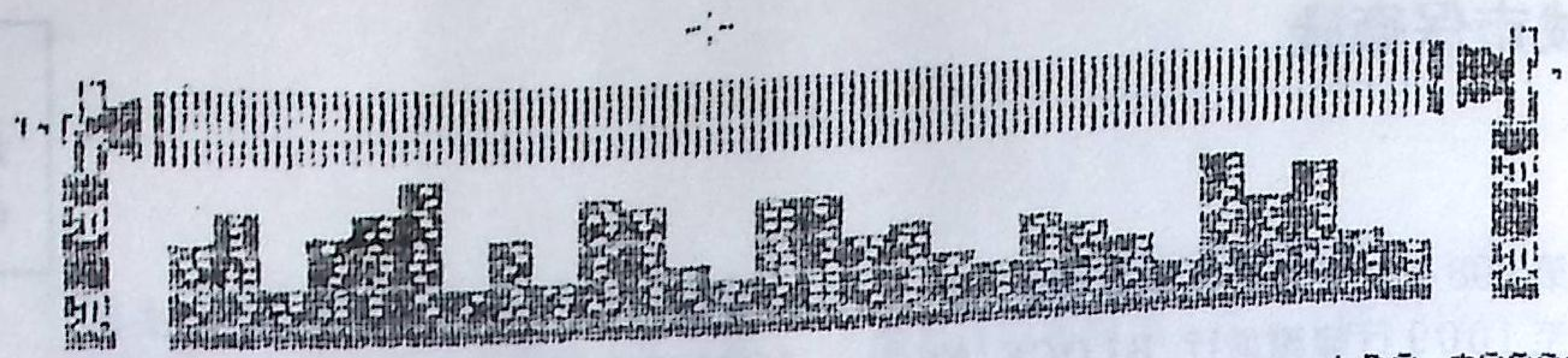
遊戲右上角顯示城市的安全狀態，藍色表示十分安全，綠色表示不甚安全，而紅色當然是隨時會完蛋了。

下次再爲各位編寫另外一個遊戲，再會。■

編者註：由於本遊戲加插有自動尋找MOCKING BOARD的SUBROUNTING，故進行遊戲時，SLOT 4 不能有任何其他咭，否則會出現沒聲或HANG 機情況。

```
GETADR1   LDA TABLO,Y
          CLC
          ADC HORIZ
          STA HIRESL1
          LDA TABHI,Y
          STA HIRESH1
          LDY #$00
          RTS
;
;
;
CHKBOUND  LDX #$00
CHKBD1    LDA AXBYTE,X
          CMP #$29
          BEQ SKIP1
          CMP #$FB
          BEQ SKIP2
CHKBD2    INX
          INX
          INX
          CPX #$8
          BLT CHKBD1
          RTS
SKIP1     LDA #$FC
          STA AXBYTE,X
          JMP CHKBD2
SKIP2     LDA #$28
          STA AXBYTE,X
          JMP CHKBD2
;
;
;
;
;
DRAWLSR   LDY VERT
          JSR GETADR
          LDY #$00
          LDA #$60
          EOR (HIRESL),Y
          STA (HIRESL),Y
          DEC VERT
          LDA VERT
          CMP #$90
          BGE DRAWLSR
DRAW.1    LDY VERT
          JSR GETADR
          LDY #$00
          LDA #$60
          STA (HIRESL),Y
          DEC VERT
          LDA VERT
          CMP MKVERT1
          BGE DRAW.1
          RTS
;
;
XDRAWLSR  LDY VERT
          JSR GETADR
          LDY #$00
          LDA #$60
          EOR (HIRESL),Y
          STA (HIRESL),Y
          DEC VERT
          LDA VERT
          CMP #$90
          BGE XDRAWLSR
XDRAW.1   LDY VERT
          JSR GETADR
          LDY #$00
          LDA #$00
          STA (HIRESL),Y
          DEC VERT
          LDA VERT
          CMP MKVERT1
          BGE XDRAW.1
          RTS
```

城市保衛戰



LISTING#2

```
         BEQ  ABNEW
CHKAB2   RTS
ABNEW    LDY  NUM
         LDA  AXBYTE,X
         CLC
         ADC  #$01
         STA  ABX,Y
         STA  XBYTE
         LDA  AXBIT,X
         STA  ABB,Y
         STA  XBIT
         LDA  AVERT,X
         CLC
         ADC  #$0C
         STA  ABY,Y
         LDY  NUM
         STA  AFLAG,Y
         STA  V
         JMP  XAB4
MOVEAB   LDY  NUM
         LDA  ABX,Y
         STA  XBYTE
         LDA  ABB,Y
         STA  XBIT
         LDA  ABY,Y
         STA  V
         CLC
         ADC  SPEED
         CMP  #$91
         BLT  EXIST
         CMP  #$C2
         BLT  EXIST1
         CMP  #$D0
         BGE  EXIST
         JMP  XAB2.1
XAB1     LDA  #$00
         LDY  NUM
         STA  AFLAG,Y
         JMP  XAB2
EXIST    LDY  NUM
         STA  ABY,Y
         JSR  $8FA2
         LDY  NUM
         LDA  ABY,Y
         STA  V
         JMP  XAB4
EXIST1   LDY  NUM
         STA  ABY,Y
         JSR  DRAW
EXIST2   LDY  NUM
         LDA  ABY,Y
         STA  V
         JSR  DRAW
         LDA  COLLSN
         BEQ  XAB2
         LDY  NUM
         LDA  ABY,Y
         CMP  #$BD
         BGE  XAB
         CMP  #$A0
         BGE  XAB3
         JSR  $8FA2
XAB2.1   LDA  WAVE
         BEQ  XAB2.2
         DEC  MAX
XAB2.2   JMP  XAB1
XAB2     RTS
XAB      JSR  DRAW
         JMP  XAB2.1
XAB4     JSR  $8F00
         RTS
XAB3     JSR  $8FA2
         LDY  NUM
         LDA  ABX,Y
         STA  XBYTE
         LDA  ABB,Y
         STA  XBIT
         LDA  ABY,Y
         CLC
         ADC  BDIST
         STA  V
         LDA  #$D
         STA  SHNUM
         JSR  DRAW
         DEC  SAFETY
         LDA  SAFETY
         CMP  #$82
         BEQ  XAB5
         CMP  #$46
         BEQ  XAB6
         JMP  XAB7
XAB5     JSR  GREENBLK
         JMP  XAB7
XAB6     JSR  REDBCK
XAB7     LDA  #$A
         STA  EPCOUNT
         LDA  #$40
         STA  EPPITCH
XAB3.A   STA  $C056
         INC  EPPITCH
         INC  EPPITCH
         LDA  EPPITCH
         STA  PITCH
         LDA  #$2
         STA  TIME
         JSR  SOUND
         INC  SPEAKER
         STA  $C057
         INC  EPPITCH
         INC  EPPITCH
         INC  EPPITCH
         INC  EPPITCH
         LDA  EPPITCH
         STA  PITCH
         LDA  #$3
         STA  TIME
         JSR  SOUND
         DEC  SPEAKER
         DEC  EPCOUNT
         BNE  XAB3.A
         JMP  XAB2.1
;
DRAW     LDA  #$00
         STA  COLLSN
         LDA  SHNUM
         ASL
         TAY
         LDA  $9380,Y
         STA  $08
         INY
         LDA  $9380,Y
         STA  $09
         LDA  XBIT
         ASL
         TAY
         LDA  ($08),Y
         TAX
         INY
         LDA  ($08),Y
         CLC
         ADC  $08
         STA  $08
         TXA
         ADC  $09
         STA  $09
         LDY  #$01
LOOP.A   LDA  ($08),Y
         STA  $20F9,Y
         DEY
         BPL  LOOP.A
         LDA  $08
         STA  LOOP.C+1
         STA  LOOP.B+1
         LDA  $09
         STA  LOOP.C+2
         STA  LOOP.B+2
         LDX  #$02
         LDA  V
         STA  $0A
LOOP.G   LDA  $20F9
         STA  $05
         LDY  V
         CPY  #$C0
         BCC  LOOP.D
         LDY  #$80
         BCS  LOOP.E
LOOP.D   LDA  $92C0,Y
         STA  $06
         LDA  $9200,Y
         STA  $07
         LDY  XBYTE
LOOP.E   CPY  #$28
         BCS  LOOP.F
LOOP.B   LDA  $7000,X
         AND  ($06),Y
         AND  #$7F
         BEQ  LOOP.C
         STA  COLLSN
LOOP.C   LDA  $7000,X
         EOR  ($06),Y
         STA  ($06),Y
LOOP.F   INX
         INY
         DEC  $05
         BNE  LOOP.E
         INC  V
         DEC  $20FA
         BNE  LOOP.G
         LDA  $0A
         STA  V
         RTS
```

最後一個立體打井遊戲

陳德來

打井遊戲相信讀者們必定玩過，立體打井各位亦有不少版本，但我相信這是最後一個的立體打井了，往後必定沒有一個打井遊戲可以同這個比較。小點心思也玩不過你的對手——電腦。

這遊戲由16條柱（4×4）格式組成。每個玩者都可選定一種的色彩來將圓環放置在柱上。每條柱最多可放4個圓環。遊戲開始時電腦會詢問你選定顏色。

遊戲的目的是取得直線一排的四個圓環，無論是橫、直、平面對角或立體對角、或直柱。請參看圖㈠。

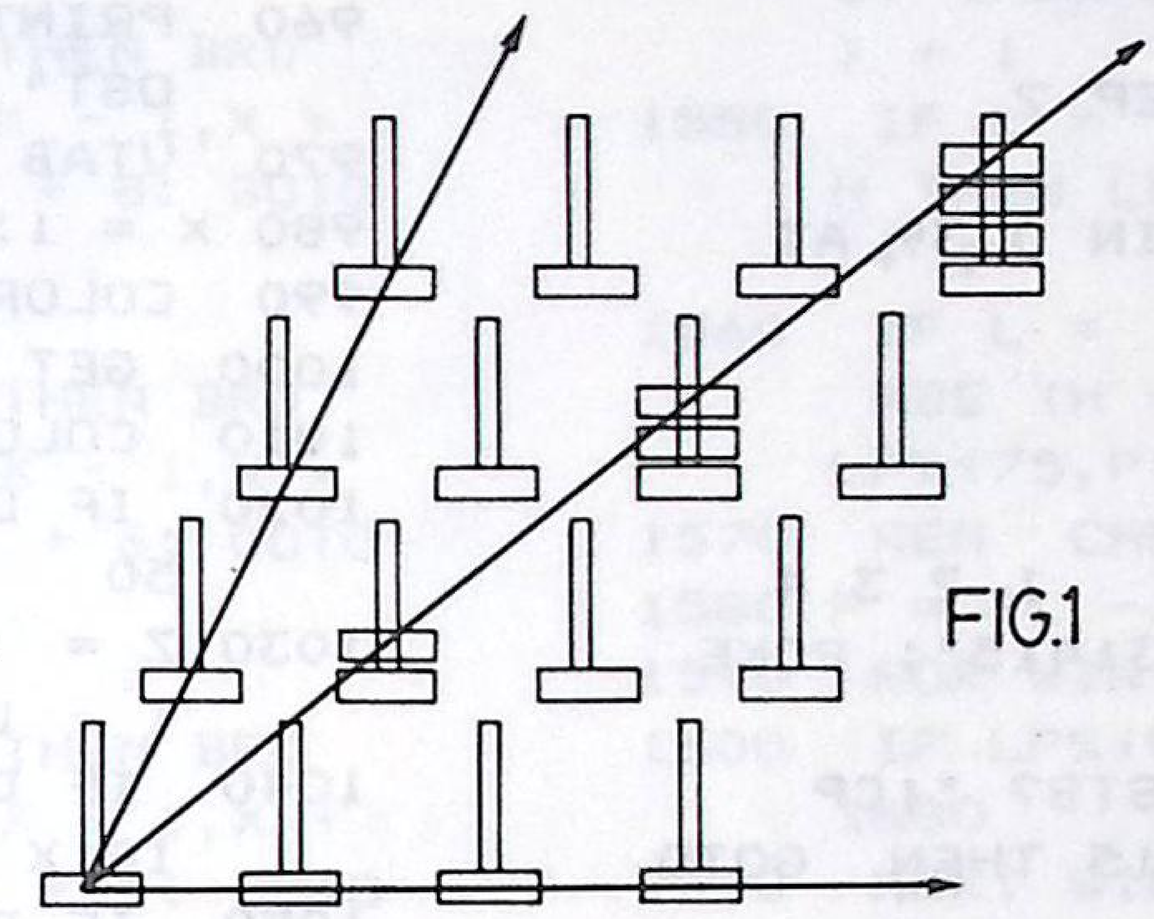


FIG.1

遊戲最好能用彩色電視進行。否則，如果是綠色螢幕的便要選擇有強烈色差的顏色，同時要說明有多少個玩家。（0代表電腦玩，1代表你和電腦鬥，2代表2個玩家進行遊戲）。假如跟電腦玩，則有5個難度供選擇，數目越大，難度越高。難度5對你來說贏電腦的機會幾乎是等於零。

所有柱及圓環是使用低解像圖頁來顯示，利用I、J、K、M鍵來控制放圓環的位置，X鍵代表選定柱。電腦會自動計分。■

```
10  REM ***************************
12  REM **                       **
15  REM ** THREE-D FOUR SCORE **
18  REM **    BY  COMPUHOUSE   **
20  REM ** COPYRIGHT(C)  1985 **
22  REM **                       **
25  REM ***************************
80  REM  INITIALIZATION
90  GOTO 2240
100  REM : SUBROUTINE TO CHECK FO
   R OPENS & CLEARS IN A LINE
110 YP = 0:YC = 0
120  FOR Z = 0 TO 9 STEP 3
130 WP = LM%(LINE,Z):LP = LM%(LIN
   E,Z + 1):HP = LM%(LINE,Z + 2
   )
140 FM = WP + (LP * 4)
150  IF OPEN(FM) = HP THEN YP = Y
   P + 1:W = WP:L = LP:H = HP: GOTO
   170
160  IF BRD%(LP,WP,HP) = 0 THEN Y
   C = YC + 1
170  NEXT Z
180  RETURN
190  REM : SUBROUTINE TO FIND NM,
   NP & NC IN A LINE
200 YP = 0:YC = 0
210  FOR LINE = 0 TO 75
220  IF LP%(LINE,P) = NM THEN  GOTO
   260
230  NEXT LINE
240 LINE = LINE - 1
250  RETURN
260  GOSUB 100
270  IF YP = NP AND YC = (4 - NM)
   - YP THEN  RETURN
280 YP = 0:YC = 0: GOTO 230
290  REM : START OF COMPUTER'S RE
   SPONSE
300 P = PL - 1
310  IF DF% = 1 THEN  GOTO 450
320  IF DF% = 2 THEN  GOTO 410
330 NM = 3:NP = 1: GOSUB 190
340  IF YP = 1 THEN  GOTO 570
350  IF P = 0 THEN P = 1: GOTO 37
   0
360 P = 0
370  GOSUB 190
380 P = PL - 1: IF YP = 1 THEN  GOTO
   570
390  IF DF% = 3 THEN  GOTO 450
400  IF DF% = 4 THEN  GOTO 430
410 NM = 2:NP = 2: GOSUB 190
420  IF YP = 2 THEN  GOTO 570
430 NM = 2:NP = 1: GOSUB 190
440  IF YP = 1 AND YC = 1 THEN  GOTO
   570
450 RC = 0
460 RC = RC + 1
470  IF RC = 15 THEN  GOTO 520
480 RW =  RND (1):RL =  RND (1)
490 W =  INT (RW * 4):L =  INT (R
   L * 4)
500  IF BRD%(L,W,3) = 0 THEN  GOTO
   570
510  GOTO 460
520  FOR W = 0 TO 3
530  FOR L = 0 TO 3
540  IF BRD%(L,W,3) = 0 THEN  GOTO
   570
550  NEXT L
560  NEXT W
```

```
570 X = (W * 8 + 1) - ((L - 3) *
     4):Y = L * 10: GOTO 1250
580  GR : SPEED= 255:S1 = 0:S2 =
     0
590 C = 1
600  FOR X = 5 TO 33 STEP 2
610  COLOR= C
620  VLIN 0,39 AT X: VLIN 0,39 AT
     X + 1
630 C = C + 1
640  NEXT X
650  HOME
660  VTAB 21: PRINT "     1 2 3 4
      5 6 7 8 9 101112131415": POKE
     34,21
670  INPUT "COLOR OF POSTS? ";CP
680  IF CP < 1 OR CP > 15 THEN  GOTO
     670
690  INPUT "COLOR FOR PLAYER #1?
     ";C1
700  IF C1 < 1 OR C1 > 15 THEN  GOTO
     690
710  INPUT "COLOR FOR PLAYER #2?
     ";C2
720  IF C2 < 1 OR C2 > 15 THEN  GOTO
     710
730  INPUT "NUMBER OF PLAYERS? ";
     PN
740  IF PN < 0 OR PN > 2 GOTO 730
750  IF PN = 2 GOTO 780
760  INPUT "SELECT DIFFICULTY 1-5
     : ";DF%
770  IF DF% < 1 OR DF% > 5 THEN  GOTO
     760
780  POKE 34,0:S1 = 0:S2 = 0
790  FOR A = 0 TO 3
800  FOR B = 0 TO 3
810  FOR X = 0 TO 3
820 BRD%(A,B,X) = 0
830  NEXT X: NEXT B: NEXT A
840  FOR A = 0 TO 15
850 OPEN(A) = 0
860  NEXT A
870 PL = 1
880  GR : HOME : COLOR= CP
890  VLIN 2,9 AT 13: VLIN 2,9 AT
     21: VLIN 2,9 AT 29: VLIN 2,9
      AT 37: VLIN 12,19 AT 9: VLIN
     12,19 AT 17: VLIN 12,19 AT 2
     5: VLIN 12,19 AT 33
900  VLIN 22,29 AT 5: VLIN 22,29 AT
     13: VLIN 22,29 AT 21: VLIN 2
     2,29 AT 29: VLIN 32,39 AT 1:
      VLIN 32,39 AT 9: VLIN 32,39
      AT 17: VLIN 32,39 AT 25
910  FOR A = 0 TO 75
920 LP%(A,0) = 0:LP%(A,1) = 0
930  NEXT A
940  IF PN = 0 THEN  HOME : GOTO
     290
950  HOME : PRINT "I = UP, K = RI
     GHT, J = LEFT, M = DOWN
960  PRINT "X = PLACE MARKER ON P
     OST"
970  VTAB 24: PRINT "PLAYER # 1"
980 X = 1:Y = 30:L = 3:W = 0
990  COLOR= 15: PLOT X,Y
1000  GET DIR$
1010  COLOR= 0: PLOT X,Y
1020  IF DIR$ = "X" THEN  GOTO 12
     50
1030 Z =  PEEK ( - 16336):Z =  PEEK
     ( - 16336)
1040  IF DIR$ = "I" THEN Y = Y -
     10:X = X + 4:L = L - 1
1050  IF DIR$ = "K" THEN X = X +
     8:W = W + 1
1060  IF DIR$ = "J" THEN X = X -
     8:W = W - 1
1070  IF DIR$ = "M" THEN Y = Y +
     10:X = X - 4:L = L + 1
1080  IF X < 1 AND Y = 30 THEN X =
     25:L = 3:W = 3
1090  IF X < 5 AND Y = 20 THEN X =
     29:L = 2:W = 3
1100  IF X < 9 AND Y = 10 THEN X =
     33:L = 1:W = 3
1110  IF X < 13 AND Y = 0 THEN X =
     37:L = 0:W = 3
1120  IF X > 25 AND Y = 30 THEN X
     = 1:L = 3:W = 0
1130  IF X > 29 AND Y = 20 THEN X
     = 5:L = 2:W = 0
1140  IF X > 33 AND Y = 10 THEN X
     = 9:L = 1:W = 0
1150  IF X > 37 AND Y = 0 THEN X =
     13:L = 0:W = 0
1160  IF Y > 30 AND X =  - 3 THEN
     X = 13:Y = 0:L = 0:W = 0
1170  IF Y > 30 AND X = 5 THEN X =
     21:Y = 0:L = 0:W = 1
1180  IF Y > 30 AND X = 13 THEN X
     = 29:Y = 0:L = 0:W = 2
1190  IF Y > 30 AND X = 21 THEN X
     = 37:Y = 0:L = 0:W = 3
1200  IF Y < 0 AND X = 17 THEN X =
     1:Y = 30:L = 3:W = 0
1210  IF Y < 0 AND X = 25 THEN X =
     9:Y = 30:L = 3:W = 1
1220  IF Y < 0 AND X = 33 THEN X =
     17:Y = 30:L = 3:W = 2
1230  IF Y < 0 AND X = 41 THEN X =
     25:Y = 30:L = 3:W = 3
1240  GOTO 990
1250  IF PL = 1 THEN  COLOR= C1:P
     C = C1
1260  IF PL = 2 THEN  COLOR= C2:P
```

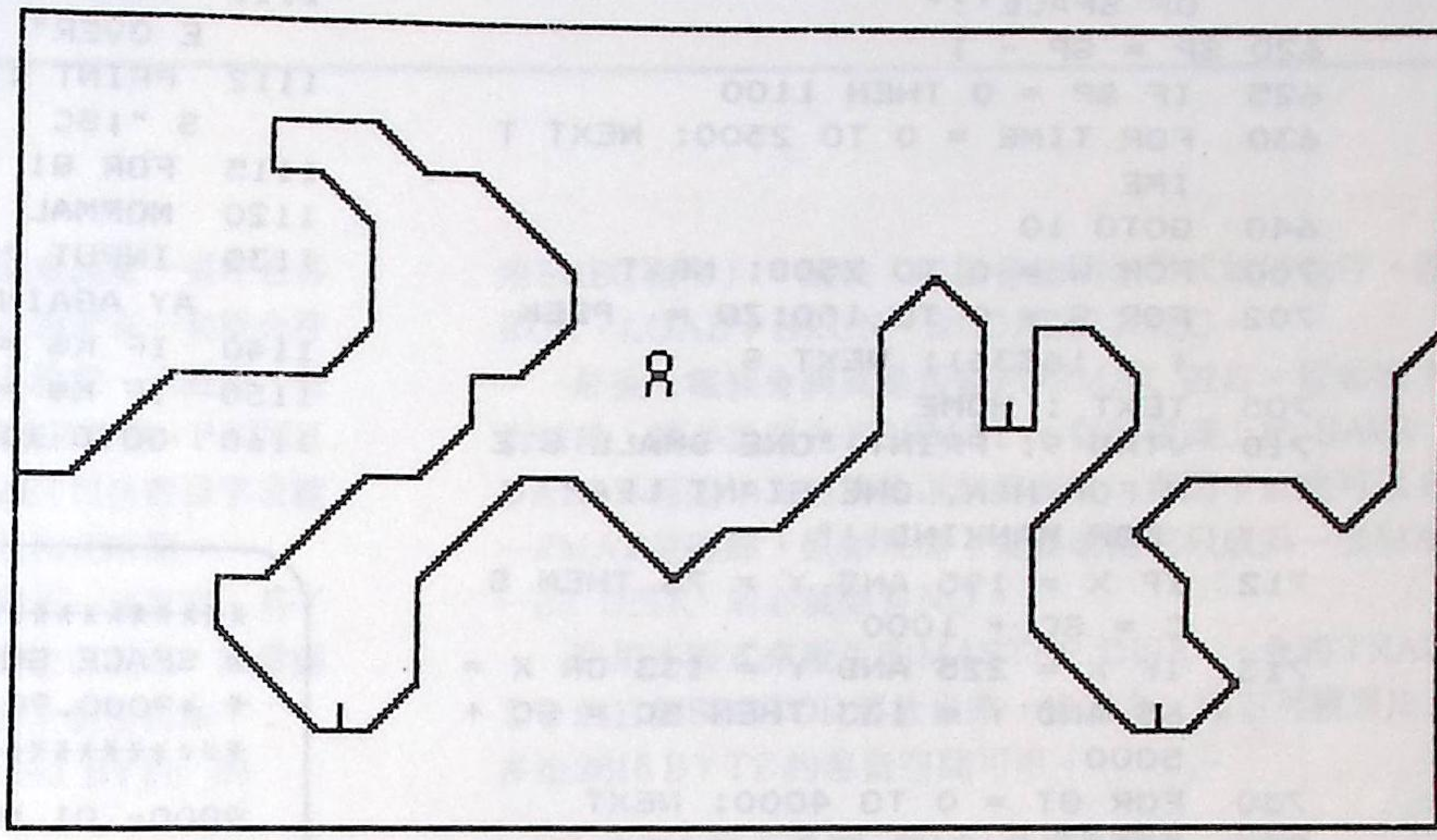
```
C = C2
1270  IF BRD%(L,W,3) <  > 0 THEN
  PRINT "POST IS FILLED!": GOTO
  990
1280 H = 0
1290  IF BRD%(L,W,H) = 0 THEN BRD
 %(L,W,H) = PL: HLIN X - 1,X +
 1 AT Y + 9: PLOT X,Y + 8: GOTO
  1360
1300 H = 1
1310  IF BRD%(L,W,H) = 0 THEN BRD
 %(L,W,H) = PL: HLIN X - 1,X +
 1 AT Y + 7: PLOT X,Y + 6: GOTO
  1360
1320 H = 2
1330  IF BRD%(L,W,H) = 0 THEN BRD
 %(L,W,H) = PL: HLIN X - 1,X +
 1 AT Y + 5: PLOT X,Y + 4: GOTO
  1360
1340 H = 3
1350  IF BRD%(L,W,H) = 0 THEN BRD
 %(L,W,H) = PL: HLIN X - 1,X +
 1 AT Y + 3: PLOT X,Y + 2
1360  REM  CONVERT W,L,H TO LINE
 # AND UPDATE
1370 FM = W + (L * 4)
1380 OPEN(FM) = H + 1
1390  REM  WIDTH FIRST
1400 P = PL - 1
1410 ZERO = L * 10:LINE = ZERO +
 H:LP%(LINE,P) = LP%(LINE,P) +
 1
1420 LINE = (ZERO + 4) + W:LP%(LI
 NE,P) = LP%(LINE,P) + 1
1430  IF W = H THEN LINE = ZERO +
 9:LP%(LINE,P) = LP%(LINE,P) +
 1
1440  IF W =  ABS (H - 3) THEN LI
 NE = ZERO + 8:LP%(LINE,P) =
 LP%(LINE,P) + 1
1450  REM  LENGTH NEXT
1460 ZERO = (W * 6) + 40:LINE = Z
 ERO + H:LP%(LINE,P) = LP%(LI
 NE,P) + 1
1470  IF L = H THEN LINE = ZERO +
 4:LP%(LINE,P) = LP%(LINE,P) +
 1
1480  IF L =  ABS (H - 3) THEN LI
 NE = ZERO + 5:LP%(LINE,P) =
 LP%(LINE,P) + 1
1490  REM  HORIZONTAL DIAGONALS N
 EXT
1500  IF L = W THEN LINE = 64 + H
 :LP%(LINE,P) = LP%(LINE,P) +
 1
1510  IF L =  ABS (W - 3) THEN LI
 NE = 68 + H:LP%(LINE,P) = LP
 %(LINE,P) + 1
1520  REM  OUTSIDE CORNER DIAGS N
 EXT
1530  IF L = W AND L = H THEN LP%
 (72,P) = LP%(72,P) + 1
1540  IF L = W AND L =  ABS (H -
 3) THEN LP%(73,P) = LP%(73,P
 ) + 1
1550  IF L =  ABS (W - 3) AND L =
 H THEN LP%(74,P) = LP%(74,P)
  + 1
1560  IF L =  ABS (W - 3) AND L =
 ABS (H - 3) THEN LP%(75,P) =
 LP%(75,P) + 1
1570  REM  CHECK FOR GAME WON
1580 P = PL - 1
1590  FOR WIN = 0 TO 75
1600  IF LP%(WIN,P) = 4 THEN  GOTO
  1680
1610  NEXT WIN
1620  IF PL = 1 THEN PL = 2: HOME
 : VTAB 23: HTAB 24: PRINT "P
 LAYER #";PL: GOTO 1640
1630 PL = 1: HOME : VTAB 23: HTAB
 6: PRINT "PLAYER #";PL
1640  IF PN = 0 THEN  PRINT "PLAY
 ER #1 = ";S1;"      PLAYER #2
  = ";S2: GOTO 290
1650  IF PN = 2 THEN  GOTO 990
1660  IF PN = 1 AND PL = 1 THEN  GOTO
 990
1670  GOTO 290
1680  REM  SOMEONE HAS WON
1690  FLASH : HOME : VTAB 22: HTAB
 12: PRINT "PLAYER #";PL;" WI
 NS": NORMAL
1700 NC = 0
1710  FOR Z1 = 0 TO 21
1720  COLOR= NC
1730  FOR Z = 0 TO 9 STEP 3
1740 WP = LM%(WIN,Z):LP = LM%(WIN
 ,Z + 1):HP = LM%(WIN,Z + 2)
1750 XX = (WP * 8) - ((LP - 3) *
 4):YY = (LP * 10) - ((HP - 3
 ) * 2) + 3
1760  HLIN XX,XX + 2 AT YY: PLOT
 XX + 1,YY - 1
1770  NEXT Z
1780  IF NC = 0 THEN NC = PC: GOTO
 1800
1790  IF NC = PC THEN NC = 0
1800  NEXT Z1
1810  IF PL = 1 THEN S1 = S1 + 1
1820  IF PL = 2 THEN S2 = S2 + 1
1830  IF PN = 1 THEN  GOTO 1920
1840  IF PN = 0 THEN  GOTO 2550
1850  VTAB 23: PRINT "PLAYER #1 =
 ";S1;"      PLAYER #2 = ";S2
1860  VTAB 24: PRINT "ANY KEY FOR
 NEXT GAME,   S TO START OVER
 "
```

```
1870  PRINT "C  TO CHANGE PLAYER
    # AND DIFFICULTY"
1880  GET G$
1890  IF G$ = "S" THEN  GOTO 580
1900  IF G$ = "C" GOTO 730
1910  GOTO 790
1920  VTAB 23: PRINT "YOUR SCORE=
      ";S1;"      MY SCORE= ";S2
1930  GOTO 1860
1940  DATA  0,0,0,1,0,0,2,0,0,3,0
     ,0,0,0,1,1,0,1,2,0,1,3,0,1,0
     ,0,2,1,0,2,2,0,2,3,0,2,0,0,3
     ,1,0,3,2,0,3,3,0,3
1950  DATA  0,0,0,0,0,1,0,0,2,0,0
     ,3,1,0,0,1,0,1,1,0,2,1,0,3,2
     ,0,0,2,0,1,2,0,2,2,0,3,3,0,0
     ,3,0,1,3,0,2,3,0,3
1960  DATA  0,0,3,1,0,2,2,0,1,3,0
     ,0,0,0,0,1,0,1,2,0,2,3,0,3
1970  DATA  0,1,0,1,1,0,2,1,0,3,1
     ,0,0,1,1,1,1,1,2,1,1,3,1,1,0
     ,1,2,1,1,2,2,1,2,3,1,2,0,1,3
     ,1,1,3,2,1,3,3,1,3
1980  DATA  0,1,0,0,1,1,0,1,2,0,1
     ,3,1,1,0,1,1,1,1,1,2,1,1,3,2
     ,1,0,2,1,1,2,1,2,2,1,3,3,1,0
     ,3,1,1,3,1,2,3,1,3
1990  DATA  0,1,3,1,1,2,2,1,1,3,1
     ,0,0,1,0,1,1,1,2,1,2,3,1,3
2000  DATA  0,2,0,1,2,0,2,2,0,3,2
     ,0,0,2,1,1,2,1,2,2,1,3,2,1,0
     ,2,2,1,2,2,2,2,2,3,2,2,0,2,3
     ,1,2,3,2,2,3,3,2,3
2010  DATA  0,2,0,0,2,1,0,2,2,0,2
     ,3,1,2,0,1,2,1,1,2,2,1,2,3,2
     ,2,0,2,2,1,2,2,2,2,2,3,3,2,0
     ,3,2,1,3,2,2,3,2,3
2020  DATA  0,2,3,1,2,2,2,2,1,3,2
     ,0,0,2,0,1,2,1,2,2,2,3,2,3
2030  DATA  0,3,0,1,3,0,2,3,0,3,3
     ,0,0,3,1,1,3,1,2,3,1,3,3,1,0
     ,3,2,1,3,2,2,3,2,3,3,2,0,3,3
     ,1,3,3,2,3,3,3,3,3
2040  DATA  0,3,0,0,3,1,0,3,2,0,3
     ,3,1,3,0,1,3,1,1,3,2,1,3,3,2
     ,3,0,2,3,1,2,3,2,2,3,3,3,3,0
     ,3,3,1,3,3,2,3,3,3
2050  DATA  0,3,3,1,3,2,2,3,1,3,3
     ,0,0,3,0,1,3,1,2,3,2,3,3,3
2060  DATA  0,0,0,0,1,0,0,2,0,0,3
     ,0,0,0,1,0,1,1,0,2,1,0,3,1,0
     ,0,2,0,1,2,0,2,2,0,3,2,0,0,3
     ,0,1,3,0,2,3,0,3,3
2070  DATA  0,0,0,0,1,1,0,2,2,0,3
     ,3,0,0,3,0,1,2,0,2,1,0,3,0
2080  DATA  1,0,0,1,1,0,1,2,0,1,3
     ,0,1,0,1,1,1,1,1,2,1,1,3,1,1
     ,0,2,1,1,2,1,2,2,1,3,2,1,0,3
     ,1,1,3,1,2,3,1,3,3
2090  DATA  1,0,0,1,1,1,1,2,2,1,3
     ,3,1,0,3,1,1,2,1,2,1,1,3,0
2100  DATA  2,0,0,2,1,0,2,2,0,2,3
     ,0,2;0,1,2,1,1,2,2,1,2,3,1,2
     ,0,2,2,1,2,2,2,2,3,2,2,0,3
     ,2,1,3,2,2,3,2,3,3
2110  DATA  2,0,0,2,1,1,2,2,2,2,3
     ,3,2,0,3,2,1,2,2,2,1,2,3,0
2120  DATA  3,0,0,3,1,0,3,2,0,3,3
     ,0,3,0,1,3,1,1,3,2,1,3,3,1,3
     ,0,2,3,1,2,3,2,2,3,3,2,3,0,3
     ,3,1,3,3,2,3,3,3,3
2130  DATA  3,0,0,3,1,1,3,2,2,3,3
     ,3,3,0,3,3,1,2,3,2,1,3,3,0
2140  DATA  3,3,0,2,2,0,1,1,0,0,0
     ,0,3,3,1,2,2,1,1,1,1,0,0,1,3
     ,3,2,2,2,2,1,1,2,0,0,2,3,3,3
     ,2,2,3,1,1,3,0,0,3
2150  DATA  0,3,0,1,2,0,2,1,0,3,0
     ,0,0,3,1,1,2,1,2,1,1,3,0,1,0
     ,3,2,1,2,2,2,1,2,3,0,2,0,3,3
     ,1,2,3,2,1,3,3,0,3
2160  DATA  3,3,3,2,2,2,1,1,1,0,0
     ,0,3,3,0,2,2,1,1,1,2,0,0,3,3
     ,0,0,2,1,1,1,2,2,0,3,3,3,0,3
     ,2,1,2,1,2,1,0,3,0
2170  FOR Z = 0 TO 75
2180  FOR Z1 = 0 TO 11
2190  READ A%
2200 LM%(Z,Z1) = A%
2210  NEXT Z1
2220  NEXT Z
2230  RETURN
2240 S1 = 0:S2 = 0:PL = 1: SPEED=
     255
2250  DIM BRD%(3,3,3),LM%(75,11),
     LP%(75,1),OPEN(15)
2260  TEXT : HOME : VTAB 5: HTAB
     13: PRINT "3-D FOUR SCORE": VTAB
     12: HTAB 14: PRINT "PLEASE W
     AIT": VTAB 22: PRINT "* COPY
     RIGHT (C) 1985 COMPUHOUSE *"
2270  GOSUB 2170
2280  REM  ABOVE SPEED SHOULD BE
     100
2290  TEXT : HOME
2540  GOTO 580
2550  VTAB 24: PRINT "ANY KEY TO
     CONTINUE, S TO START OVER"
2560  GET G$
2570  IF G$ = "S" GOTO 580
2580  GOTO 790
```

月球一小步，人類一大步

這個登月遊戲，你必定會玩得非常緊張。你要小心控制太空船降落到指定的降落地點。降落地點共有三個，左右兩個需要較高的駕駛技術，如果降落成功，可取得五千分，而中間一個降落台則很容易降落，成功者只取得一千分。

遊戲中共有4隻太空船。你可用I，J，K，M 四個鍵來控制，移動時，按任何鍵都可停止移動。本遊戲共有兩個程式，其中一個機械碼是用來繪劃太空船。



我們歡迎讀者將這遊戲改良，增加其難度及使版面多些變化。獲取用的作品可獲得任何兩期的電腦時代程式磁碟。有關投寄程式方法請參閱〔投稿需知〕。

```
0  REM  ** MOON  LANDING **
1  REM  ** BY COMPUHOUSE **
2 HS = 0: POKE 234,0
3  PRINT  CHR$ (4);"BLOAD SPACE S
    HIP"
4  POKE 232,00: POKE 233,144
5 SC = 0:SP = 3:HS = 0
7  TEXT : HOME
8  PRINT "THIS PROGRAM WAS WRITTE
    N BY COMPU HOUSE"
9  FOR PL = 0 TO 2500: NEXT PL
10  HGR : HCOLOR= 3
20  HPLOT 0,90 TO 10,90 TO 30,70 TO
    60,70 TO 70,60 TO 70,40 TO 6
    0,30 TO 50,30 TO 50,20 TO 70
    ,20 TO 80,30 TO 90,30 TO 110
    ,50 TO 110,60 TO 100,70 TO 9
    0,70 TO 90,80 TO 80,90
30  HPLOT 80,90 TO 70,90 TO 50,11
    0 TO 40,110 TO 40,120 TO 60,
    140 TO 70,140 TO 80,130 TO 8
    0,110 TO 100,90 TO 100,90 TO
    110,90 TO 130,110 TO 140,100
     TO 150,100 TO 170,80
40  HPLOT 170,80 TO 170,60 TO 180
    ,50 TO 190,60 TO 190,80 TO 2
    00,80 TO 200,60 TO 210,60 TO
    220,70 TO 220,80 TO 200,100 TO
    200,120 TO 220,140 TO 230,14
    0 TO 240,130 TO 230,120
50  HPLOT 230,120 TO 230,110 TO 2
    20,100 TO 230,90 TO 250,90 TO
    260,100 TO 270,90 TO 270,70 TO
    279,60
60  HPLOT 0,0 TO 279,0 TO 279,159
     TO 0,159 TO 0,0
65  HPLOT 195,75 TO 195,80: HPLOT
    225,135 TO 225,140: HPLOT 65
    ,135 TO 65,140
69  GET A$
70 X = 5:Y = 83
75  HOME
76  VTAB 22: PRINT " SHIPS LEFT "
    ;SP
77  VTAB 23: PRINT " SCORE ";SC
80  FOR W2 = 0 TO 1
90  ROT= 0
100  SCALE= 1
110  DRAW 1 AT X,Y
130  IF  PEEK ( - 16384) < 127 THEN
     150
140  GET A$
150  IF X < 0 THEN 600
160  IF Y < 1 THEN 600
170  IF X > 279 THEN 600
180  IF Y > 159 THEN 600
190  IF A$ = "I" THEN Y = Y - 2:
     HCOLOR= 0: DRAW 1 AT X,Y + 2:
     HCOLOR= 3: DRAW 1 AT X,Y
200  IF A$ = "M" THEN Y = Y + 2:
     HCOLOR= 0: DRAW 1 AT X,Y - 2:
     HCOLOR= 3: DRAW 1 AT X,Y
210  IF A$ = "J" THEN X = X - 2:
     HCOLOR= 0: DRAW 1 AT X + 2,Y:
     HCOLOR= 3: DRAW 1 AT X,Y
220  IF A$ = "K" THEN X = X + 2:
     HCOLOR= 0: DRAW 1 AT X - 2,Y:
     HCOLOR= 3:  DRAW 1 AT X,Y
230  IF X = 195 AND Y = 73 OR X =
     225 AND Y = 133 OR X = 65 AND
     Y = 133 THEN 700
240  IF X = 3 OR X = 275 OR Y = 3
     THEN 600
```

月球一小步，人類一大步

```
499  IF  PEEK (234) > 2 THEN 1010
500  GOTO 130
600  FOR AZS = 0 TO 1500: NEXT
605  TEXT : HOME
610  VTAB 9: HTAB 13: PRINT "OUT
     OF SPACE!!"
620 SP = SP - 1
625  IF SP = 0 THEN 1100
630  FOR TIME = 0 TO 2500: NEXT T
     IME
640  GOTO 10
700  FOR W = 0 TO 2500: NEXT
702  FOR S = 0 TO 150:ZQ =  PEEK
     ( - 16336): NEXT S
705  TEXT : HOME
710  VTAB 9: PRINT "ONE SMALL STE
     P FOR MAN, ONE GIANT LEAP
      FOR MANKIND!!"
712  IF X = 195 AND Y = 73 THEN S
     C = SC + 1000
713  IF X = 225 AND Y = 133 OR X =
     65 AND Y = 133 THEN SC = SC +
     5000
730  FOR GT = 0 TO 4000: NEXT
800  GOTO 10
1010  FOR P = 0 TO 64
1020  ROT= P: SCALE= 2
1025  DRAW 1 AT X,Y
1027 F =  PEEK ( - 16336)
1030  NEXT P
1040  FOR Q = 0 TO 2500: NEXT
1050 SP = SP - 1
1060  IF SP = 0 THEN 1100
1070  GOTO 10
1100  TEXT : HOME
1110  VTAB 9: HTAB 16: PRINT "GAM
     E OVER"
1112  PRINT : PRINT "YOUR SCORE I
     S ";SC
1115  FOR Q1 = 0 TO 1500: NEXT
1120  NORMAL
1130  INPUT "WOULD YOU LIKE TO PL
     AY AGAIN?(Y/N)  ";K$
1140  IF K$ = "Y" THEN  RUN
1150  IF K$ = "N" THEN  END
1160  GOTO 1130
```

■

```
***************
* SPACE SHIP *
* *9000.9015 *
***************

9000- 01 00 04 00 12 3F 20 64
9008- 2D 15 36 1E 07 2E 15 36
9010- DF 23 64 25 00 A9
*
```

音量控制

蘋果的喇叭輸出一直以來都是它的弱點。不過，我們依然可以利用一些程式來進行一些玩意。例如本程式就是，他可控制揚聲器的音量輸出。

本文的示範程式可以容許你來指定揚聲器的音量輸出。

在使用時，必須要留意三項事情：DURATION，PITCH及VOLUMN。對DURATION 來說，0是最長，256（$FF）是次長，1是最短。對PITCH來說，0是最低，256（$FF）是次低，1是最高。而VOLUME，最大音量是PITCH 值的一半，假如音量是0，而PITCH亦是0，或大過0，則會產生別樣的聲音，因此，小心VOLUMN值要介乎1或比PITCH 值少1的範圍內。要使用這子程式，先BLOAD VO—LUMN CONTROL入記憶體，然後POKE DURATION值入位址769，PITCH 值入位址771，及VOLUME值入788，然後CALL 768。

本文第二個程式是示範程式，以固定的PITCH，音量由細到大，而又由大到細。■

VOLUMN CONTROL

```
0300-   A0 90      LDY   #$90
0302-   A9 60      LDA   #$60
0304-   38         SEC
0305-   ED 14 03   SBC   $0314
0308-   8D 0C 03   STA   $030C
030B-   A2 30      LDX   #$30
030D-   CA         DEX
030E-   D0 FD      BNE   $030D
0310-   AD 30 C0   LDA   $C030
0313-   A2 30      LDX   #$30
0315-   CA         DEX
0316-   D0 FD      BNE   $0315
0318-   AD 30 C0   LDA   $C030
031B-   88         DEY
031C-   D0 ED      BNE   $030B
031E-   60         RTS
```

CONTROL DEMO

```
1000-   A9 01      LDA   #$01
1002-   8D 14 03   STA   $0314
1005-   20 00 03   JSR   $0300
1008-   EE 14 03   INC   $0314
100B-   AD 14 03   LDA   $0314
100E-   C9 60      CMP   #$60
1010-   D0 F3      BNE   $1005
1012-   60         RTS
```

用最簡易方便的方法造出一張隨心所欲的磁碟

磁碟度身定造器

閣下有沒有夢想過擁有一套工具能隨意設定一張符合各類情況需求的磁碟。筆者相信，雖然現在還未有一套能合符水準的工具，但閣下仍然可隨自己心意去設定一張磁碟，例如將BOOTING PROGRAM的FILENAME改變，PATCH DOS ，給予任何字母開首的FILENAME(包括數目字及標準符號）等等，但做起來時却有少許麻煩和花時間。

不過，看完本文之後，閣下的煩惱將可一掃而空。為了方便我們平常將任何一張磁碟按不同的需求而設定，筆者編寫了以下一個〔磁碟度身訂造器〕，它的功能包括如下：

① FORMAT 一張比平常儲存量多14,592 BYTE 的 DATA DISK。

② 只要一步的功夫便可INITIALIZE一張MASTER DISK。

③ 隨自己心意用任何字母組成的FILENAME ，包括數目字及標點符號。

④ 按自己要求給予新磁碟的SECTOR OFFSET。

⑤ 只要簡單的BLOAD，便可PATCH MASTER DISK上的DOS。

磁碟度身定造器（見列表）被執行時，會將磁碟上0—2磁軌的DOS 影像全部搬上主機記憶體，並放在＄1B00至＄3FFF 的位址上。主要的搬移子程序是由＄1B00 開始至＄1CFF 。而DOS本身則是由＄1D00開始至＄3FFF 。程式只適用於48K APPLE ，假如在DOS已被搬上RAM咭的情況下，執行本程式時，會出現ERROR訊息。

執行程式時，會出現一個有五項選擇的目錄表。該五項選擇包括：

```
1. MASTER DISK (CREAT/CONVERT)
2. STORAGE DISK (CREAT/CONVERT)
3. DEFAULTS (VIEW/CHANGE)
4. PATCH DOS IMAGE
5. EXIT
ENTER CHOICE ?
```

(1) MASTER DISK 製造

這項選擇設立一張MASTER DISK。在設立之前，大家應先看看選擇(3)的DEFAULT，設定各項要求。選擇(1)執行時，會先詢問你想定的BOOTING程式的FILE NAME。這個BOOTING程式可以是APPLESOFT，INTEGER BASIC，機械語言，甚至乎是SEQUENTAIL TEXT FILE（需要用EXEC指令）。隨後，電腦會詢問你程式如何執行，即是RUN，LOAD，BRUN，BLOAD或EXEC。

最後，電腦會問及是否要FORMAT 磁碟。假如閣下用新磁碟，就必須要先FORMAT 。假如磁碟上已 SAVE 有DATA，而閣下又想清除這些舊資料，則閣下依然可以FO—RMAT這磁碟，但如果閣下是要改變它只成為一張MAST—ER DISK，則必須要答NO。

利用本程式來產生的MASTER DISK ，會將TRACK ＄2 的11個SECTOR解放出來，換言之，閣下可獲得比正常多出2816 BYTE的容量空間可用。

(2) STORAGE DISK 製造

有關第二項選擇STORAGE DISK，可以幫助閣下很快的製造一張沒有DOS的STORAGE DISK，由於沒有DOS的存在，故可增加磁碟容量12032 BYTE。在未製造STOR—AGE DISK的時候，請先看看第二項選擇的DEFAULT（OPTION 3）。

執行第二項選擇時，閣下將會被問及一些BOOTING訊息。由於STORAGE DISK 沒有DOS ，有需要加上某些訊息，好讓該磁碟在BOOT起時，螢幕會有指示，或忠告訊息。

在輸入這些忠告訊息時，最多可輸入200個字母。閣下來利用ESC I、J、K及M來將CURSOR移動。

輸入完訊息後，按RET。欲想觀看所輸入的內容是否正確可按CONTROL D〔RET〕。

然後螢幕會詢問你是否要FORMAT磁碟。有關的步驟和情況跟OPTION 1的MASTER DISK 一樣。留意一項事，一張用本程式製造的STORAGE DISK並不能用來轉換為MASTER DISK ，除非重新將該碟FORMAT過。

(3) DEFAULTS 設定

至於第三項的選擇（DEFAULTS）是用來更改DISK VOLUME，VOLUME NO等的東西。當選擇這項OPTION後，螢幕會出現一個有8項選擇的目錄，如下：

```
1. SLOT..................6
2. DRIVE.................1
3. VOLUME................000
4. PERMIT ANY FILENAME...FALSE
5. CHECK TRACK ZERO......FALSE
6. CATALOG FILES (MAX)...105
7. SECTOR OFFSET.........$2
8. 'DISK VOLUME'
```

(1) SLOT及(2) DRIVE 是指示出用來工作的磁碟機及槽位。

(3) VOLUME是指待FORMAT 的磁碟的VOL. NO.。只有在INIT一張新碟時，本項改變才有效。

(4) PERMIT ANY FILENAME是用來PATCH DOS，容許用英文字母以外的數字、標點來作FILE NAME。FAL-SE表示不允許，而TRUE表示可以用任何字符作FILEN-AME，除了0（，），（SPACE），及RETURN鍵。

(5) CHECK TRACK 0是用來檢查TRACK 0。一般來說STORAGE DISK 是可利用到TRACK 0，故毋須CHECK，而本項選擇亦因此只適用於MASTER。TRUE 表示進行PATCH。

(6) CATALOG FILE。TRACK $11 是用來放FILE CATALOG，未修改過的磁碟可容許最多至105個FILENA-ME，假如閣下認爲沒有需要用太多的FILENAME，大可利用本項功能來達成。由於每個SECTOR可放7個FILE-NAME，假如你只需要35個FILENAME，則你可以將餘下的10個SECTOR（2560 BYTE）騰空出來。這項功能必須在FORMAT一張新碟情況下始能生效。

(7) SECTOR OFFSET。對於任何的STORAGE DISK，大家都可在每條TRACK上OFFSET，而在MASTER DISK 上則只能在TRACK 0至2。一般來說數值9是最快LOAD或BLOAD磁碟資料。新的SECTOR OFFSET只在FORMAT 一張新碟情況下有效。

(8) DISK VOLUME 這項選擇是可讓大家將DISK VO-LUME 更改至任何字樣。

(4)輕易的PATCH DOS

第四項PATCH DOS IMAGE 選擇，可將你帶入MO-NITOR，進行PATCH DOS。記住DOS的內容是放在$1D00至$3FFF 的位址上。你可以直接BLOAD 有關的PATCH內容入這位址上。這項選擇是可以提供一個快捷又簡單的工具，而毋須用ZAP或ADU等程式。

將修改好後，按RETURN並再用OPTION(1) 將新PA-TCH過的DOS 放入磁碟上。

各位千祈要緊記，進行PATCH時，不要用有珍貴資料的磁碟，應該用一張工作碟。執行第1至3項選擇時，如要中途離開返回主程式，可 按Q。■

```
************************
**  DISK CUSTOMIZER   **
** COPYRIGHT 1985 BY **
**   COMPUTING AGE    **
************************
*803.193E

0803- 4C 51 0E 02 11
0808- 11 00 01 00 00 00 00 00
0810- 0F 01 60 01 00 00 00 22
0818- 08 00 00 00 00 00 00 00
0820- 60 01 00 01 EF D8 A9 A9
0828- 8D 92 32 A2 30 AD 0A 08
0830- F0 02 A2 B0 8E 4B 20 AD
0838- 0B 08 D0 12 A9 AD 8D 05
0840- 31 A9 BB 8D 06 31 A9 35
0848- 8D 07 31 4C 7A 08 AD 0F
0850- 08 18 69 31 8D B3 08 8D
0858- B9 08 69 04 8D B0 08 69
0860- 07 8D A7 08 A2 02 BD A5
0868- 08 9D 05 31 CA 10 F7 A2
0870- 11 BD A8 08 9D E0 3C CA
0878- 10 F7 A2 0F 8A 9D BA 08
0880- CA 30 21 38 ED 06 08 B0
0888- F4 69 10 8E 0D 08 A0 0F
0890- D9 BA 08 F0 08 88 CC 0D
0898- 08 F0 E2 D0 F3 38 E9 01
08A0- B0 EC 90 E5 60 4C E0 3C
08A8- C8 B1 42 D0 07 88 AD BB
08B0- 35 4C 08 31 88 B1 42 4C
08B8- 14 31 00 0D 0B 09 07 05
08C0- 03 01 0E 0C 0A 08 06 04
08C8- 02 0F 20 00 0B C6 CF D2
08D0- CD C1 D4 D4 C9 CE C7 A0
08D8- C4 C9 D3 CB AE A0 D7 C1
08E0- C9 D4 A0 AE AE AE 8D 00
08E8- AD 0F 08 18 69 3F 85 07
08F0- 8D 60 09 A9 21 85 06 8D
08F8- 5F 09 A0 03 B9 42 09 91
0900- 06 88 10 F8 20 DC 03 84
0908- 06 85 07 A0 00 A9 0B 91
0910- 06 C8 AD D2 03 91 06 A0
0918- 04 AD 09 08 91 06 C8 AD
0920- 13 08 91 06 C8 AD 12 08
0928- 4A 4A 4A 4A 91 06 A0 0C
0930- A9 DF 91 06 C8 A9 19 91
0938- 06 20 D6 03 90 03 4C E2
0940- 0A 60 EA 20 46 09 8E 0D
0948- 08 A5 44 C9 03 90 29 C9
0950- 22 D0 11 A5 19 C9 0E D0
0958- 0B A2 03 BD 78 09 9D 21
0960- BF CA 10 F7 A5 3F D0 04
0968- A9 FF 85 19 E6 19 A6 19
0970- BD BA 08 85 3F AE 0D 08
0978- E6 3F A5 3F 60 8D 1D 08
0980- A5 06 8D 19 08 A5 07 8D
0988- 1A 08 A9 08 A0 11 20 D9
0990- 03 90 03 4C E2 0A 60 E6
0998- 06 D0 02 E6 07 60 E6 00
09A0- D0 02 E6 01 60 AD 07 08
09A8- 8D 15 08 A2 10 8E 0D 08
09B0- CE 0D 08 AD 0D 08 C9 01
09B8- F0 13 8D 16 08 20 D9 0A
09C0- A9 01 20 7D 09 AD E0 19
09C8- D0 E6 4C 18 0A AD 10 08
09D0- 49 0F 18 69 01 8D 16 08
09D8- 20 D9 0A A9 01 20 7D 09
09E0- AD DE 1A F0 03 4C 18 0A
09E8- A9 00 8D E0 19 8D E1 19
09F0- 20 D9 0A A9 02 20 7D 09
09F8- A9 01 8D 0C 08 20 58 0A
0A00- AC 10 08 B9 3A 0A 8D 5B
0A08- 1A B9 49 0A 8D 5C 1A A9
0A10- 02 8D 0C 08 20 58 0A 60
0A18- 20 03 0B C3 CF D5 CC C4
0A20- CE A7 D4 A0 C6 D2 C5 C5
0A28- A0 C3 C1 D4 C1 CC CF C7
0A30- A0 D4 D2 C1 C3 CB AE 8D
0A38- 00 60 00 7F 3F 1F 0F 07
0A40- 03 01 00 00 00 00 00 00
0A48- 00 00 FE FE FE FE FE FE
0A50- FE FE 7E 3E 1E 0E 06 02
0A58- AD 07 08 8D 15 08 A9 00
0A60- 8D 16 08 20 D9 0A AD 0C
0A68- 08 20 7D 09 90 03 4C E2
0A70- 0A AD 0C 08 C9 01 D0 07
0A78- AD DF 19 C9 04 D0 01 60
0A80- 20 2F FB 20 00 0B 87 D6
0A88- CF CC D5 CD C5 A0 D4 C1
0A90- C2 CC C5 A0 CF C6 A0 C3
0A98- CF CE D4 C5 CE D4 D3 A0
0AA0- CE CF D4 A0 C6 CF D5 CE
0AA8- C4 A0 CF CE 8D 8D D4 D2
0AB0- C1 C3 CB A0 CE D5 CD C2
0AB8- C5 D2 A0 A4 00 AD 07 08
0AC0- 20 DA FD 20 03 0B 8D 8D
0AC8- C6 C1 D4 C1 CC A0 C1 C2
0AD0- CF D2 D4 AE 8D 00 4C 32
0AD8- 0E A9 DF 85 06 A9 19 85
0AE0- 07 60 20 2F FB 20 00 0B
0AE8- 87 C1 C2 CF D2 D4 C5 C4
0AF0- BA C4 C9 D3 CB A0 C5 D2
0AF8- D2 CF D2 8D 00 4C 32 0E
0B00- 20 58 FC 68 85 00 68 85
0B08- 01 20 9E 09 A0 00 B1 00
0B10- F0 09 20 ED FD 20 9E 09
0B18- 4C 0E 0B 20 9E 09 6C 00
0B20- 00 A9 AD D0 02 A9 AA A2
0B28- 25 20 ED FD CA 10 FA A9
0B30- 8D 20 ED FD 60 A9 AE 20
0B38- ED FD A5 24 C9 1D D0 F5
0B40- 60 F0 09 20 03 0B D4 D2
0B48- D5 C5 00 60 20 03 0B C6
0B50- C1 CC D3 C5 00 60 85 44
```

磁碟度身定造器

```
0B58- AD 0F 08 18 69 2E 85 07
0B60- A9 42 85 06 6C 06 00 EE
0B68- 16 08 AD 16 08 C9 10 D0
0B70- 08 EE 15 08 A9 00 8D 16
0B78- 08 60 A0 43 A2 0B BD 89
0B80- 0B 99 DF 19 88 CA 10 F6
0B88- 60 FF FE 00 00 FF FF 00
0B90- 00 FF FF 00 00 A0 41 B9
0B98- DF 19 C9 00 D0 0B A9 E0
0BA0- 99 DF 19 A9 FF 88 99 DF
0BA8- 19 60 A9 00 85 06 A9 36
0BB0- 85 07 A9 00 8D 15 08 8D
0BB8- 16 08 AD 0C 08 20 7D 09
0BC0- 20 67 0B E6 07 AD 16 08
0BC8- C9 0A D0 EE A9 00 85 06
0BD0- A9 1B 85 07 AD 0C 08 20
0BD8- 7D 09 20 67 0B E6 07 AD
0BE0- 16 08 C9 05 D0 EE AD 15
0BE8- 08 C9 02 D0 E7 A9 01 8D
0BF0- 0E 08 60 20 6F FD A0 00
0BF8- 84 3E E0 02 B0 06 20 A7
0C00- FF A5 3E 60 20 2D 0C B0
0C08- 02 85 3E 20 2D 0C B0 07
0C10- A8 B9 3A 0C 20 25 0C CA
0C18- 30 10 20 2E 0C B0 0B 29
0C20- 03 A8 B9 44 0C 18 65 3E
0C28- 85 3E A5 3E 60 CA BD 00
0C30- 02 38 E9 B0 9D 00 02 C9
0C38- 10 60 00 0A 14 1E 28 32
0C40- 3C 46 50 5A 00 64 C8 00
0C48- A9 A4 20 ED FD 20 6F FD
0C50- A0 00 20 A7 FF A5 3E 60
0C58- 20 6F FD A0 00 B9 00 02
0C60- C9 D1 F0 03 C9 8D 60 4C
0C68- 51 0E 8D 10 C0 AD 00 C0
0C70- 10 FB 8D 10 C0 C9 D1 F0
0C78- 03 C9 8D 60 4C 51 0E A5
0C80- 24 8D BA 0C A5 25 8D BB
0C88- 0C 20 03 0B A8 D9 AF CE
0C90- A9 A0 00 20 58 0C C9 D9
0C98- F0 0A C9 8D F0 06 C9 CE
0CA0- D0 03 C9 FF 60 AD BA 0C
0CA8- 85 24 AD BB 0C 85 25 20
0CB0- 5B FB A9 87 20 ED FD 4C
0CB8- 89 0C 00 00 20 03 0B A0
0CC0- A0 A0 A0 B1 AE A0 D3 CC
0CC8- CF D4 00 20 35 0B AD 12
0CD0- 08 4A 4A 4A 4A 18 69 B0
0CD8- 20 ED FD 20 03 0B 8D A0
0CE0- A0 A0 A0 B2 AE A0 C4 D2
0CE8- C9 D6 C5 00 20 35 0B AD
0CF0- 13 08 18 69 B0 20 ED FD
0CF8- A9 8D 20 ED FD 60 20 00
0D00- 0B C6 CF D2 CD C1 D4 A0
0D08- D4 C8 C9 D3 A0 C4 C9 D3
0D10- CB BF 8D 8D D7 C1 D2 CE
0D18- C9 CE C7 BA A0 D4 C8 C9
0D20- D3 A0 D0 D2 CF C3 C5 C4
0D28- D5 D2 C5 A0 D7 C9 CC CC
0D30- 8D C9 CE C9 D4 C9 C1 CC
0D38- C9 DA C5 A0 D4 C8 C5 A0
0D40- C4 C9 D3 CB A0 C1 CE C4
0D48- A0 C5 D2 C1 D3 C5 A0 C1
0D50- CC CC 8D C4 C1 D4 C1 A0
0D58- C6 D2 CF CD A0 C9 D4 AE
0D60- 8D 8D C6 CF D2 CD C1 D4
0D68- BF A0 00 20 7F 0C 08 C9
0D70- 8D D0 04 68 68 68 60 20
0D78- 58 FC 28 08 D0 24 20 03
0D80- 0B 2A 2A 20 04 09 13 0B
0D88- 20 17 09 0C 0C 20 02 05
0D90- 20 09 0E 09 14 09 01 0C
0D98- 09 1A 05 04 20 2A 2A 8D
0DA0- 8D 00 20 03 0B D0 CC C1
0DA8- C3 C5 A0 C4 C9 D3 CB A0
0DB0- D4 CF A0 C2 C5 A0 00 28
0DB8- 08 F0 10 20 03 0B 03 0F
0DC0- 0E 16 05 12 14 05 04 00
0DC8- 4C D6 0D 20 03 0B 03 12
0DD0- 05 01 14 05 04 00 20 03
0DD8- 0B A0 C9 CE D4 CF BA 8D
0DE0- 8D 00 20 BC 0C 20 03 0B
0DE8- 8D DB D1 DD A0 C5 D8 C9
0DF0- D4 D3 AE A0 A0 DB D2 C5
0DF8- D4 D5 D2 CE DD A0 C2 C1
0E00- C3 CB D3 A0 D5 D0 AE 8D
0E08- 8D D0 D2 CF C3 C5 C5 C4
0E10- BF A0 00 20 7F 0C D0 04
0E18- C9 8D D0 04 68 4C FE 0C
0E20- 28 D0 03 4C CA 08 60 20
0E28- 03 0B 8D 8D C4 CF CE C5
0E30- AE 00 20 03 0B 8D 8D C1
0E38- CE D9 A0 CB C5 D9 A0 D4
0E40- CF A0 C3 CF CE D4 C9 CE
0E48- D5 C5 AE AE AE 00 20 6A
0E50- 0C A9 00 8D 14 08 A9 03
0E58- 8D 4C 09 A9 BA 85 33 20
0E60- 89 FE 20 93 FE AD D2 03
0E68- 38 E9 1D C9 9E 90 03 4C
0E70- 68 10 C9 5D 90 F9 8D 0F
0E78- 08 AD D1 03 C9 BF D0 EF
0E80- 20 E3 03 84 06 85 07 A0
0E88- 01 B1 06 8D 12 08 8D 20
0E90- 08 C8 B1 06 8D 13 08 8D
0E98- 21 08 AD 0F 08 18 69 2C
0EA0- 85 07 A0 01 84 06 88 B1
0EA8- 06 8D 07 08 8D 08 A9
0EB0- 51 8D F2 03 A9 0E 8D F3
0EB8- 03 49 A5 8D F4 03 20 2F
0EC0- FB 20 58 FC 20 25 0B 20
0EC8- 03 0B AA A0 A0 A0 A0 A0
0ED0- A0 A0 A0 A0 A0 A0 A0 A0
0ED8- A0 A0 A0 A0 A0 A0 A0 A0
0EE0- A0 A0 A0 A0 A0 A0 A0 A0
0EE8- A0 A0 A0 A0 A0 A0 A0 AA
0EF0- 8D AA A0 A0 A0 A0 A0 A0
0EF8- A0 A0 A0 A0 C4 C9 D3 CB
0F00- A0 C3 D5 D3 D4 CF CD C9
0F08- DA C5 D2 A0 A0 A0 A0 A0
0F10- A0 A0 A0 A0 A0 A0 AA 8D
0F18- AA A0 A0 A0 A0 A0 A0 A0
0F20- A0 A0 A0 A0 A0 A0 A0 A0
0F28- A0 A0 A0 A0 A0 A0 A0 A0
0F30- A0 A0 A0 A0 A0 A0 A0 A0
0F38- A0 A0 A0 A0 A0 AA 8D AA
0F40- A0 C3 CF D0 D9 D2 C9 C7
0F48- C8 D4 A0 B1 B9 B8 B4 A0
0F50- C2 D9 A0 C3 CF CD D0 D5
0F58- D4 C9 CE C7 A0 C1 C7 C5
0F60- A0 A0 A0 A0 AA 8D AA A0
0F68- A0 A0 A0 A0 A0 A0 A0 A0
0F70- A0 A0 A0 A0 A0 A0 A0 A0
0F78- A0 A0 A0 A0 A0 A0 A0 A0
0F80- A0 A0 A0 A0 A0 A0 A0 A0
0F88- A0 A0 A0 AA 8D 00 20 25
0F90- 0B A9 09 85 22 AD 0E 08
0F98- F0 03 4C 89 10 20 00 0B
0FA0- CC CF C1 C4 C9 CE C7 A0
0FA8- CD C1 D3 D4 C5 D2 A0 C9
0FB0- CD C1 C7 C5 A0 C9 CE D4
0FB8- CF A0 CD C5 CD CF D2 D9
0FC0- 8D 00 A9 01 8D 0C 08 20
0FC8- AA 0B AD 00 1B F0 26 AD
0FD0- 01 36 C9 A5 D0 1F A2 00
0FD8- AD 4B 20 C9 30 F0 02 A2
0FE0- 01 8E 0A 08 A2 00 AD 05
0FE8- 31 C9 4C D0 02 A2 01 8E
0FF0- 0B 08 4C 89 10 20 2F FB
0FF8- 20 00 0B 8D 8D 8D CE CF
1000- D4 A0 C1 A0 CD C1 D3 D4
1008- C5 D2 A0 C4 C9 D3 CB AE
1010- 00 20 03 0B 87 8D 8D C4
1018- C9 D3 CB A0 C3 D5 D3 D4
1020- CF CD C9 DA C5 D2 A0 D3
1028- C8 CF D5 CC C4 A0 C2 C5
1030- A0 C2 D2 D5 CE A0 C6 D2
1038- CF CD 8D C1 A0 C6 D2 C5
1040- D3 C8 CC D9 A0 C2 CF CF
1048- D4 C5 C4 A0 CD C1 D3 D4
1050- C5 D2 A0 C4 C9 D3 CB 8D
1058- D4 CF A0 CF D0 C5 D2 C1
1060- D4 C5 AE 8D 00 4C FE 11
1068- 20 2F FB 20 00 0B D5 CE
1070- D5 D3 C1 C2 CC C5 A0 C4
1078- CF D3 A0 C9 CE A0 CD C5
1080- CD CF D2 D9 AE 00 4C 11
1088- 10 20 00 0B B1 AE A0 CD
1090- C1 D3 D4 C5 D2 A0 C4 C9
1098- D3 CB A0 A8 C3 D2 C5 C1
10A0- D4 C5 AF C3 CF CE D6 C5
10A8- D2 D4 A9 8D 8D B2 AE A0
10B0- D3 D4 CF D2 C1 C7 C5 A0
10B8- C4 C9 D3 CB A0 A8 C3 D2
10C0- C5 C1 D4 C5 AF C3 CF CE
10C8- D6 C5 D2 D4 A9 8D 8D B3
10D0- AE A0 C4 C5 C6 C1 D5 CC
10D8- D4 D3 A0 A8 D6 C9 C5 D7
10E0- AF C3 C8 C1 CE C7 C5 A9
10E8- 8D 8D B4 AE A0 D0 C1 D4
10F0- C3 C8 A0 C4 CF D3 A0 C9
10F8- CD C1 C7 C5 8D 8D B5 AE
1100- A0 C5 D8 C9 D4 8D 8D C5
1108- CE D4 C5 D2 A0 C3 C8 CF
1110- C9 C3 C5 BA A0 00 20 58
1118- 0C 38 E9 B1 90 04 C9 05
1120- 90 03 4C 89 10 0A A8 C8
1128- B9 32 11 48 88 B9 32 11
1130- 48 60 15 12 6D 17 B8 13
1138- 3B 11 FD 11 A9 51 8D F9
1140- 03 A9 0E 8D FA 03 20 2F
1148- FB 20 00 0B 87 C1 CE A0
1150- C9 CD C1 C7 C5 A0 CF C6
1158- A0 C4 CF D3 A0 CE CF D7
1160- A0 D2 C5 D3 C9 C4 C5 D3
1168- A0 C9 CE 8D CD C5 CD CF
1170- D2 D9 A0 C6 D2 CF CD A0
1178- A4 B1 C4 B0 B0 A0 D4 CF
1180- A0 A4 B3 C6 C6 C6 AE A0
1188- D5 D3 C5 8D D4 C8 C5 A0
1190- CD CF CE C9 D4 CF D2 A0
1198- D4 CF A0 D0 C1 D4 C3 C8
11A0- A0 D4 C8 C9 D3 A0 C9 CD
11A8- C1 C7 C5 AE 8D 8D C3 D4
11B0- D2 CC AD D9 BC D2 C5 D4
11B8- D5 D2 CE BE A0 D4 CF A0
11C0- D2 C5 C5 CE D4 C5 D2 A0
11C8- C3 D5 D3 D4 CF CD C9 DA
11D0- C5 D2 AE 8D 8D C4 CF D3
11D8- A0 C9 D3 A0 D3 D4 C9 CC
11E0- CC A0 C1 C3 D4 C9 D6 C5
11E8- CC D9 A0 C3 CF CE CE C5
11F0- C3 D4 C5 C4 AE 8D 8D 00
11F8- 20 EA 03 4C 69 FF A9 03
1200- 8D F2 03 A9 E0 8D F3 03
1208- 49 A5 8D F4 03 20 2F FB
1210- 20 EA 03 4C D0 03 20 00
1218- 0B D3 D0 C5 C3 C9 C6 D9
1220- A0 C6 C9 CC C5 CE C1 CD
1228- C5 A0 CF C6 A0 C2 CF CF
1230- D4 A0 D0 D2 CF C7 D2 C1
1238- CD AE 8D 8D C6 C9 CC C5
1240- A0 CD C1 D9 A0 C2 C5 A0
```

磁碟度身定造器

```
1248- D4 C5 D8 D4 AC A0 C2 C9
1250- CE C1 D2 D9 AC A0 C9 CE
1258- D4 C5 C7 C5 D2 AC 8D CF
1260- D2 A0 C1 D0 D0 CC C5 D3
1268- CF C6 D4 A0 D4 D9 D0 C5
1270- A0 C6 C9 CC C5 AE 8D 8D
1278- C6 C9 CC C5 CE C1 CD C5
1280- BA 00 20 6F FD E0 00 D0
1288- 03 4C 51 0E 20 95 20 A2
1290- 00 BD 00 02 C9 8D F0 08
1298- 9D 75 2A E8 E0 1D D0 F1
12A0- 20 00 0B C6 C9 CC C5 CE
12A8- C1 CD C5 BA A0 00 A2 00
12B0- BD 75 2A 20 ED FD E8 E0
12B8- 1D D0 F5 20 03 0B 8D 8D
12C0- C1 C3 D4 C9 CF CE A0 D4
12C8- CF A0 C2 C5 A0 D4 C1 CB
12D0- C5 CE A0 D7 C8 C5 CE A0
12D8- D4 C8 C9 D3 8D C4 C9 D3
12E0- CB A0 C9 D3 A0 C2 CF CF
12E8- D4 C5 C4 BA 8D 8D A0 A0
12F0- A0 A0 A0 A0 A0 A0 B1 AE
12F8- A0 D2 D5 CE 8D A0 A0 A0
1300- A0 A0 A0 A0 A0 B2 AE A0
1308- CC CF C1 C4 8D A0 A0 A0
1310- A0 A0 A0 A0 A0 B3 AE A0
1318- C2 D2 D5 CE 8D A0 A0 A0
1320- A0 A0 A0 A0 A0 B4 AE A0
1328- C2 CC CF C1 C4 8D A0 A0
1330- A0 A0 A0 A0 A0 A0 B5 AE
1338- A0 C5 D8 C5 C3 8D 00 20
1340- 03 0B 8D D7 C8 C9 C3 C8
1348- BF A0 00 20 58 0C D0 03
1350- 4C 16 12 38 E9 B1 90 E7
1358- C9 06 B0 E3 A8 B9 B4 13
1360- 8D 42 1E 20 26 08 A9 12
1368- 48 A9 9F 48 20 FE 0C 68
1370- 68 20 00 0B D0 CC C1 C3
1378- C9 CE C7 A0 C4 CF D3 A0
1380- CF CE D4 CF A0 C4 C9 D3
1388- CB AE 8D 00 A9 02 8D 0C
1390- 08 20 AA 0B A9 01 8D 0C
1398- 08 20 58 0A 20 95 0B A9
13A0- 02 8D 0C 08 20 58 0A AD
13A8- 10 08 C9 0F F0 03 20 A5
13B0- 09 4C 27 0E 06 02 34 32
13B8- 14 20 00 0B C3 D5 D2 D2
13C0- C5 CE D4 A0 C4 C5 C6 C1
13C8- D5 CC D4 A0 D6 C1 CC D5
13D0- C5 D3 BA 8D 8D 00 20 BC
13D8- 0C 20 03 0B A0 A0 A0 A0
13E0- B3 AE A0 D6 CF CC D5 CD
13E8- C5 00 20 35 0B AD 09 08
13F0- 20 56 0B 20 03 0B 8D A0
13F8- A0 A0 A0 B4 AE A0 D0 C5
1400- D2 CD C9 D4 A0 C1 CE D9
1408- A0 C6 C9 CC C5 CE C1 CD
1410- C5 00 20 35 0B AD 0A 08
1418- 20 41 0B 20 03 0B 8D A0
1420- A0 A0 A0 B5 AE A0 C3 C8
1428- C5 C3 CB A0 D4 D2 C1 C3
1430- CB A0 DA C5 D2 CF 00 20
1438- 35 0B AD 0B 08 20 41 0B
1440- 20 03 0B 8D A0 A0 A0 A0
1448- B6 AE A0 C3 C1 D4 C1 CC
1450- CF C7 A0 C6 C9 CC C5 D3
1458- A0 A8 CD C1 D8 A9 00 20
1460- 35 0B AD 10 08 0A 0A 0A
1468- 38 ED 10 08 20 56 0B 20
1470- 03 0B 8D A0 A0 A0 A0 B7
1478- AE A0 D3 C5 C3 D4 CF D2
1480- A0 CF C6 C6 D3 C5 D4 00
1488- 20 35 0B A9 A4 20 ED FD
1490- AD 06 08 18 69 B0 20 ED
1498- FD 20 03 0B 8D A0 A0 A0
14A0- A0 B8 AE A0 A7 00 A2 0B
14A8- BD AF 33 20 ED FD CA 10
14B0- F7 A9 A7 20 ED FD 20 03
14B8- 0B 8D 8D D7 C8 C9 C3 C8
14C0- BF A0 00 20 58 0C D0 03
14C8- 4C 89 10 38 E9 B1 B0 03
14D0- 4C B9 13 C9 08 B0 F9 0A
14D8- A8 C8 B9 E4 14 48 88 B9
14E0- E4 14 48 60 F3 14 24 15
14E8- 52 15 7B 15 EA 15 75 16
14F0- D0 16 01 17 20 00 0B C5
14F8- CE D4 C5 D2 A0 D3 CC CF
1500- D4 A0 CE D5 CD C2 C5 D2
1508- A0 A8 B1 AD B7 A9 A0 00
1510- 20 F3 0B C9 02 90 0B C9
1518- 08 B0 07 0A 0A 0A 0A 8D
1520- 12 08 4C B9 13 20 00 0B
1528- C5 CE D4 C5 D2 A0 C4 D2
1530- C9 D6 C5 A0 CE D5 CD C2
1538- C5 D2 A0 A8 B1 AD B2 A9
1540- A0 00 20 F3 0B C9 01 90
1548- 07 C9 03 B0 03 8D 13 08
1550- 4C B9 13 20 00 0B C5 CE
1558- D4 C5 D2 A0 D6 CF CC D5
1560- CD C5 A0 CE D5 CD C2 C5
1568- D2 A0 A8 B0 AD B2 B5 B5
1570- A9 A0 00 20 F3 0B 8D 09
1578- 08 4C B9 13 20 00 0B D4
1580- D2 D5 C5 A0 C1 CC CC CF
1588- D7 D3 A0 C6 C9 CC C5 CE
1590- C1 CD C5 D3 A0 D4 CF A0
1598- C2 C5 C7 C9 CE 8D D7 C9
15A0- D4 C8 A0 C1 CE D9 A0 CC
15A8- C5 D4 D4 C5 D2 AC A0 CE
15B0- D5 CD C2 C5 D2 AC A0 CF
15B8- D2 8D D3 D9 CD C2 CF CC
15C0- AE 8D 8D C4 CF A0 D9 CF
15C8- D5 A0 D7 C9 D3 C8 A0 C9
15D0- D4 A0 D4 D2 D5 C5 BF A0
15D8- 00 A2 01 20 7F 0C 20 63
15E0- 17 F0 02 A2 00 8E 0A 08
15E8- 4C B9 13 20 00 0B D4 D2
15F0- D5 C5 A0 CD CF C4 C9 C6
15F8- C9 C5 D3 A0 C4 CF D3 A0
1600- C9 CD C1 C7 C5 A0 D4 CF
1608- 8D C1 CC CC CF D7 A0 C4
1610- CF D3 A0 D4 CF A0 D2 C5
1618- C1 C4 A0 D4 C8 C5 A0 D3
1620- D4 CF D2 C1 C7 C5 8D D3
1628- D0 C1 C3 C5 A0 CF CE A0
1630- D4 D2 C1 C3 CB A0 DA C5
1638- D2 CF A0 CF C6 A0 D3 D4
1640- CF D2 C1 C7 C5 8D C4 C9
1648- D3 CB D3 AE 8D 8D C4 CF
1650- A0 D9 CF D5 A0 D7 C9 D3
1658- C8 A0 C9 D4 A0 D4 D2 D5
1660- C5 BF A0 00 A2 01 20 7F
1668- 0C 20 63 17 F0 02 A2 00
1670- 8E 0B 08 4C B9 13 20 00
1678- 0B C1 CE D4 C9 C3 C9 D0
1680- C1 D4 C5 C4 A0 CD C1 D8
1688- C9 CD D5 CD A0 CE D5 CD
1690- C2 C5 D2 A0 CF C6 8D C6
1698- C9 CC C5 D3 A0 C9 CE A0
16A0- C3 C1 D4 C1 CC CF C7 A0
16A8- A8 B1 B0 B5 A0 CD C1 D8
16B0- A9 BA A0 00 20 F3 0B A0
16B8- 00 38 E9 07 90 03 C8 D0
16C0- F9 C0 10 B0 09 C0 01 B0
16C8- 02 A0 01 8C 10 08 4C B9
16D0- 13 20 00 0B C5 CE D4 C5
16D8- D2 A0 D3 C5 C3 D4 CF D2
16E0- A0 CF C6 C6 D3 C5 D4 A0
16E8- A8 A4 B2 AD A4 C5 A9 A0
16F0- 00 20 48 0C C9 0F B0 07
16F8- C9 02 90 03 8D 06 08 4C
1700- B9 13 20 00 0B C5 CE D4
1708- C5 D2 A0 CE C5 D7 A0 C3
1710- C1 D4 C1 CC CF C7 A0 C8
1718- C5 C1 C4 C5 D2 A0 D0 C8
1720- D2 C1 D3 C5 AE 8D 8D DB
1728- D2 C5 D4 D5 D2 CE DD A0
1730- C5 D8 C9 D4 D3 AE 8D 8D
1738- D0 C8 D2 C1 D3 C5 BA A0
1740- 00 20 6F FD E0 00 F0 18
1748- A9 A0 9D 00 02 E8 E0 0D
1750- 90 F8 A2 0B A0 00 B9 00
1758- 02 9D AF 33 C8 CA 10 F6
1760- 4C B9 13 08 C9 8D D0 04
1768- 28 4C B9 13 28 60 20 00
1770- 0B C5 CE D4 C5 D2 A0 CD
1778- C5 D3 D3 C1 C7 C5 A0 D4
1780- CF A0 C2 C5 A0 D0 D2 C9
1788- CE D4 C5 C4 A0 D7 C8 C5
1790- CE 8D D4 C8 C9 D3 A0 D3
1798- D4 CF D2 C1 C7 C5 A0 C4
17A0- C9 D3 CB A0 C9 D3 A0 C2
17A8- CF CF D4 C5 C4 AE 8D 8D
17B0- C2 C5 C7 C9 CE A0 CE C5
17B8- D7 CC C9 CE C5 A0 D7 C9
17C0- D4 C8 A0 C3 CE D4 CC AD
17C8- C4 A0 DB D2 C5 D4 D5 D2
17D0- CE DD 8D D4 CF A0 C5 D8
17D8- C9 D4 AE A0 A0 A8 B2 B0
17E0- B0 A9 A0 C3 C8 C1 D2 C1
17E8- C3 D4 C5 D2 D3 A0 CD C1
17F0- D8 AE 8D 8D 00 20 A5 18
17F8- A9 0F 20 5B FB A9 00 85
1800- 24 8D 0D 08 20 6F FD AE
1808- 0D 08 A0 00 B9 00 02 C9
1810- 84 F0 12 9D 1C 19 E8 E0
1818- C9 F0 0A 8E 0D 08 C8 C9
1820- 8D D0 E9 F0 DF A9 00 AE
1828- 0D 08 9D 1C 19 20 58 FC
1830- 20 A5 18 20 21 0B 20 03
1838- 0B 8D 8D CF CB C1 D9 BF
1840- A0 00 20 7F 0C D0 04 C9
1848- 8D D0 03 4C 6E 17 20 26
1850- 08 A9 00 8D 4C 09 A9 18
1858- 48 A9 2C 48 20 FE 0C 68
1860- 68 A9 01 8D 0C 08 20 58
1868- 0A A2 38 BD DF 19 F0 03
1870- 4C B6 18 E8 E0 44 D0 F3
1878- 20 7A 0B A9 02 8D 0C 08
1880- 20 58 0A 20 58 FC A9 FD
1888- 85 06 A9 18 85 07 A9 00
1890- 8D 15 08 A9 02 20 7D 09
1898- AD 10 08 C9 0F F0 03 20
18A0- A5 09 4C 27 0E 20 21 0B
18A8- A2 00 BD 1C 19 F0 06 20
18B0- ED FD E8 D0 F5 60 20 00
18B8- 0B 87 C4 C9 D3 CB A0 C1
18C0- CC D2 C5 C1 C4 D9 A0 C3
18C8- CF CE D6 C5 D2 D4 C5 C4
18D0- AE 8D 8D D0 CC C1 C3 C5
18D8- A0 CE C5 D7 A0 CD C5 D3
18E0- D3 C1 C7 C5 A0 C1 CE D9
18E8- D7 C1 D9 BF A0 00 20 7F
18F0- 0C D0 07 C9 8D F0 03 4C
18F8- 83 18 4C 51 0E 01 20 93
1900- FE 20 89 FE 20 58 FC A6
1908- 2B BD 88 C0 A2 00 BD 1F
1910- 08 F0 06 20 ED FD E8 D0
1918- F5 4C 00 E0 D3 D4 CF D2
1920- C1 C7 C5 A0 C4 C9 D3 CB
1928- A0 A8 CE CF A0 C4 CF D3
1930- A9 8D 8D D3 D5 C2 CA C5
1938- C3 D4 BA A0 8D 00 00
```

合初學者程度的工具

DOS BOSS 使用詳談

達仔

相信大家讀者們對DOS BOSS 這軟件再不會感到陌生，因爲DOS BOSS 已經面世了好幾年。但因不少的初學者對DOS BOSS 還是在苦苦的摸索之中，而有關的參考書又不是這麽容易的在市面中購買，所以爲着初學者方便，現詳述DOS BOSS 的用法。

用途廣泛：

它能夠將原有的

(1) DOS COMMAND
(2) ERROR MESSAGE
(3) CATALOG FORMATTER
(4) VOLUME HEADING
(5) FILE CODE

等隨着讀者們喜歡的去更改。

使用辦法：

首先用DOS 3.3去BOOT起DOS BOSS 之磁碟，然後在衆多之檔案之中去RUN DOS BOSS。幾秒鐘之後，便可以見到DOS BOSS MENU，然後可以隨着讀者喜歡去選擇其功能。

功能詳述：

C）DOS COMMAND CHANGES

當進入了DOS COMMAND CHANGES之後，便會顯示APPLE之中的二十八個DOS指令，而每一個DOS指令都有一個不同的英文字母代表，然後按入需要更改指令的代號，再輸入新的指令，便可以完成更改指令的程序。兼且同時還會顯示原有及已更改指令的數目。此外還有兩個功能，(1)“$”是代表將所有新的DOS 指令轉回舊有的版本。(2)“M”是代表完成所需要的程序而返回DOS BOSS MENU。

E）ERROR MESSAGE CHANGES

其功能同DOS COMMAND CHANGES沒有大的分別，只是DOS COMMAND CHANGES是去更改DOS 指令，而ERROR MESSAGE CHANGES 是去更改錯誤號指令。

X）CATALOG FORMATTER

這一個功能對讀者們就更加有用，因爲每期磁碟月刊及電腦時代磁碟之檔案多至數十個，你未必可以一次的看到所有之檔案，而這功能就可以帮到大家。它能在一次之中顯示多至八十八個檔案。其功能有：

0——觀察其目錄所制定格式之形式。
1——將目錄返回原有，即一行及每行三十個字表示。
2——將目錄分爲兩行及每行十三字表示，這可以一次之中顯示四十四個檔案。
3——這和“2”一樣，不過省去了檔案前之別類。
4——將檔案分爲四行及每行九個字表示。這可以在一次之中顯示八十八個檔案。
M——返回DOS BOSS MENU。

V）VOLUME HEADING CHANGES

這功能是將原的DISK VOLUME 254 隨着大家喜歡的文字去更改，例如：可以更改爲COMPUTING AGE 14 等。其功能有：

#——是加上或減去VOLUME NO.
C——去更改新的VOLUME HEADING
M——返回DOS BOSS MENU。

在去更改VOLUME HEADING時，還有多個附加功能，包括：

@——是代表RETURN鍵，是開一新行。
^——是代表開一新行及下一行的第一個字是跟隨着上一行最尾的字母。
<——是代表向後移一格。
>——是代表移右一格。

F）FILE CODE CHANGES

這功能是去更改檔案別類之代號，其功能有：

A——是代表要更改／APPLESOFT檔案之代號。
I——是代表要更改INTEGER檔案之代號。
B——是代表要更改BINARY檔案之代號。
T——是代表要更改TEXT檔案之代號。
L——是代表要更改已鎖上檔案之代號。
U——是代表要更改沒有鎖上檔案之代號。
M——返回DOS BOSS MENU。

S）SAVING DOS CHANGES

這功能是儲存已經更改之指令到別的新磁碟之中，其功能有：

M——返回DOS BOSS MENU。
Q——這是去SET—UP 新的磁碟。首先按入“Q”，然後用“NEW”的指令洗去DOS BOSS，跟着輸入“HELLO”，最後便INIT 到別的新磁碟之中，便告完成。
P——CREAT POKE FILE
這功能是去輸入自己需要之POKE VALUE，然後再INIT到別的新磁碟之中。

這個由BEAGLE BROS 所出品之DOS BOSS，如果大家讀者們掌握其功能，眞是受用不少。■

ENTERTAINMENT

最新遊戲軟件評價

LUCIFERS REACM

軟件名稱：LUCIFER REACM
軟件類型：ADVENTURE
評　　分：＊＊＊＊



LUCIFERS REACM 是一個極高難度的ADVENTURE GAME，雖然非常難玩，但它的圖像及聲响非常引人入勝，這個故事說你因爲得了重病而死去，但你不幸落了地獄，而你要與魔鬼鬥爭，但非常多關阻止你。

由於這個ADVENTURE GAME出了不久，筆者未能一時之間想出破解之法，但筆者千辛萬苦找到關於這遊戲的KEY WORD，這些KEY WORD 有時只要獨立輸入，但有時要連同一個名詞一齊輸入，現寫出KEY WORD 給大家參考：

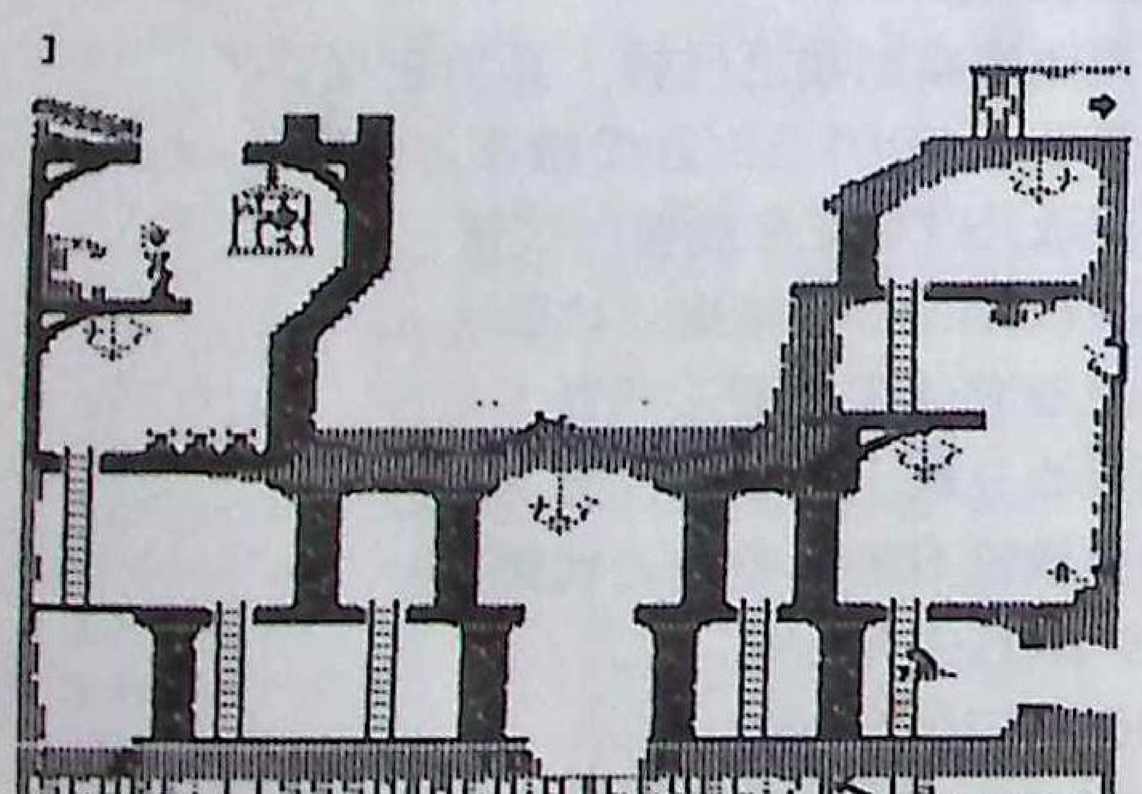
LOOK、KILL、CLIMB、READ、TALK、OPEN、UNLOCK、EAT、MOVE、DIG、DRINK、TASTE、JUMP、PUSH、THROW、SMELL、LISTEN、GRAPHIC （能夠用顯示圖像方式進行遊戲，或以TEXT畫面進行遊戲）、SAVE GAME、LOAD GAME、QUIT、PULL。

希望登出了這些KEY WORD 後，對大家玩這個ADVENTURE GAME有帮助，如筆者能破解的話，一定第一時間和大家分享。■

編者註：歡迎讀者寄來破解答案。

CONAN孫悟空

遊戲名稱：CONAN
遊戲類型：ACTION GAME
評　　分：＊＊＊＊



另一個較爲突出的遊戲，便是CO—NAN。CONAN 是一個非常具刺激性的遊戲，每一個LEVEL 都有非常難度性，現在逐一介紹：

LEVEL 1：這LEVEL 是最易玩的。只要避開雀鳥的襲擊，以後到箭咀所指的地方便行。

LEVEL 2：這LEVEL只是較爲複雜些，只要拿到鎖匙，然後去開鎖，再入去山洞便行。

LEVEL 3：這LEVEL 看上去並不難玩，但是非常煩人，你要穿過隧道，以後拿起寶石，但你千萬不要快離開，因爲筆者發覺只要在雀鳥來的時候，碰到它，便能夠獎一個人，當所有步驟完成後，便把寶石放在寶石座上，此時，水面便會升起肥皂泡，你只要跳上去，然後跳入山崖便行。

LEVEL 4：這LEVEL 是給玩者作補充之用，只要把二顆寶石放在寶石座上，便會有鎖匙降下來，拿了鎖匙後，便到門口開鎖，跟着到箭咀所指的地方便行。在這排中，只是多了些水母，只要小心些便行。

LEVEL 5：這LEVEL 是要對付噴火龍，只要殺下六條噴火龍，便會有一度門打開，衝過門，放下寶石，然後拿起鎖匙，再開另一度門便行。

LEVEL 6：這LEVEL 是筆者認爲最困難的。因爲要避過火、閃電及電眼的襲擊。只要把小電眼消滅便行，每消滅一次左上角便會出現一些樓梯，直至把整條樓梯顯現出來爲止。跟着爬上樓梯，然後把大電眼上的吊燈切斷便行，然後到箭咀所指的地方。

LEVEL 7：這LEVEL 連筆者也不能破到，但我知道是怎樣玩法，首先用劍把飛鏢射落，當射落後，便會出現寶石，然後把寶石放在寶石座上，直至把三個寶石座放滿寶石爲止。

過了第七關後，便會出現一幅美麗的圖案，圖案是什麽，筆者在此賣一個關子，留待讀者去玩！■

KARATEKA（武士道）

遊戲名稱：KARATEKA
遊戲類型：ACTION GAME
評　分：＊＊＊＊

今期爲各位介紹幾個現時最流行的遊戲。首先，爲大家介紹的是KARAT—EKA（武士道），武士道是一個ACT—ION ADVENTURE GAME。每一個LEVEL 都帶給玩者非常高的娛樂性。

遊戲開始時，你會從山崖爬上來，而當你走遠些的時候，你便會遇到一個把守大門口的高手，你只要用你的拳及腿便輕易打倒他，跟着你只要把遊戲桿推上右上角，便能夠健步如飛，再跟着，便是應接二連三的敵人，直至你進入大屋的門口爲止。

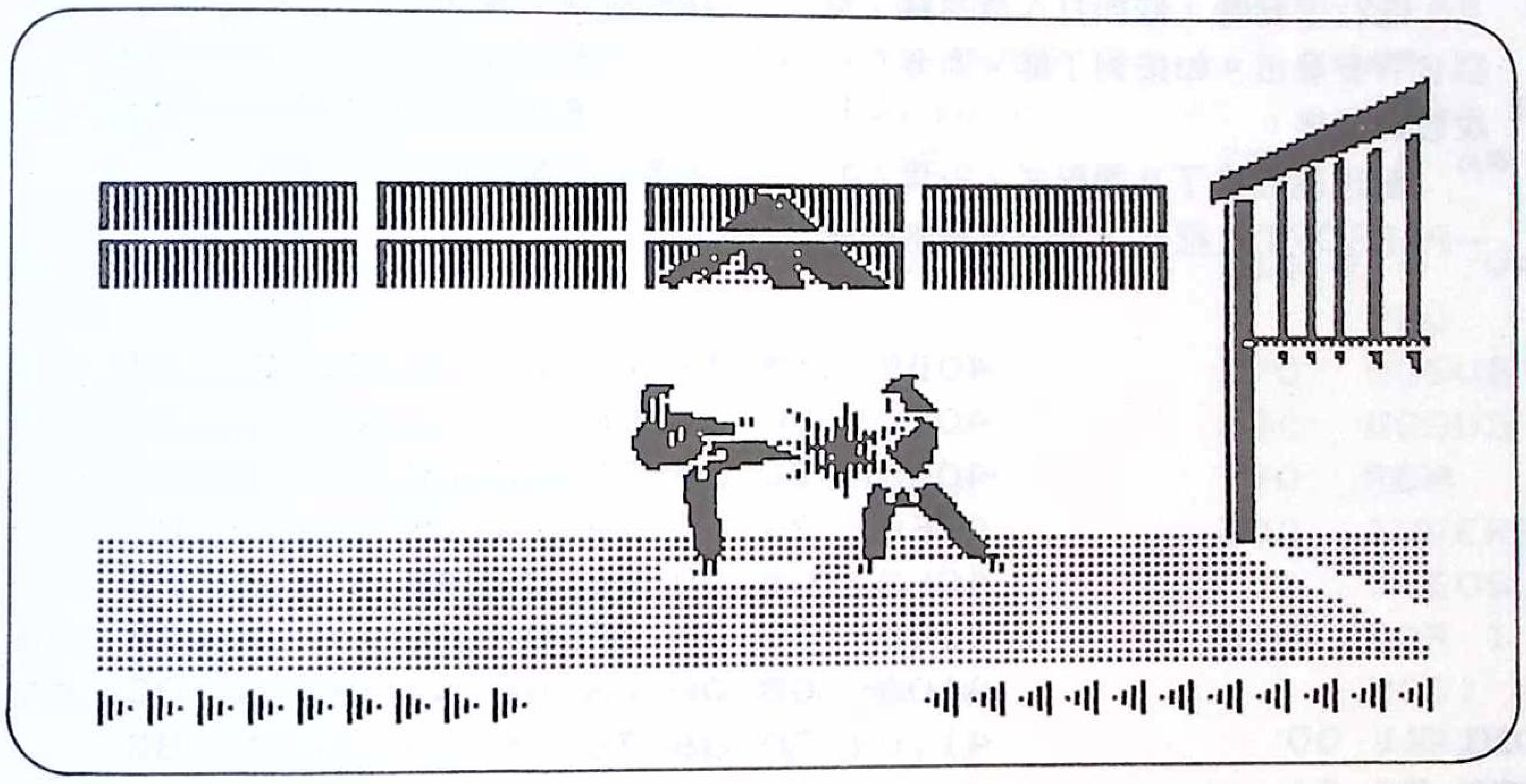
此時，LEVEL 2便開始，形式和LEVEL 1無異，只是背景不同和加上獵鷹的襲擊，而LEVEL 2的門口和LEVEL 1的門口分別就很大，因爲LEVEL 2的門口是會隨時落下的，但只要位置適中，便能夠過關了。

LEVEL 3是以房間形式做背景，每個在房裏的人都是高手，你只要盡量埋身，然後以拳和腿夾擊，便能夠擊對方，在LEVEL 3之中，最厲害的便是獵鷹，你一定要企定才能夠對付牠，大約踢牠四脚，便能夠把鷹粉碎了。

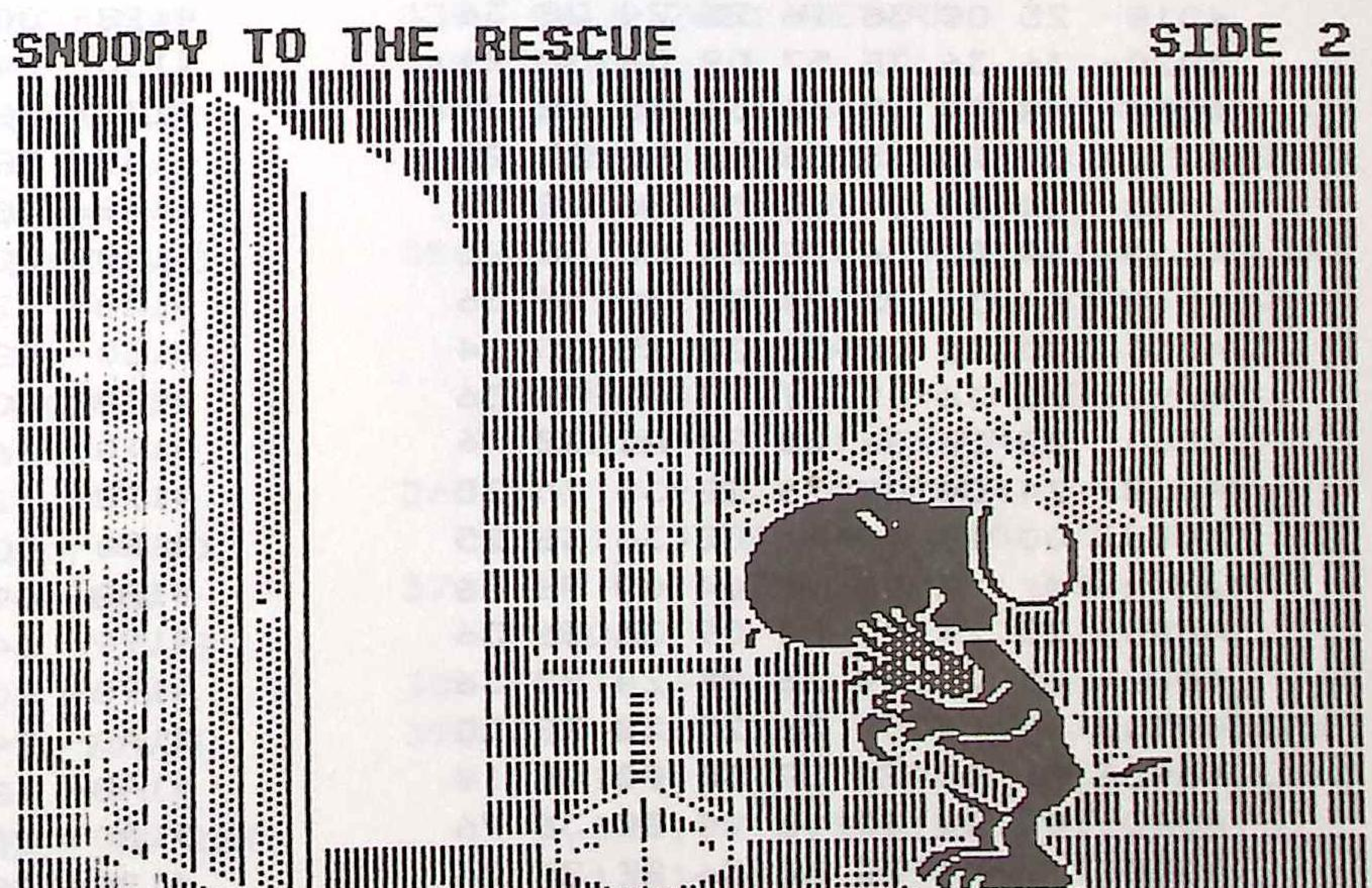
最後一個LEVEL 便是最精彩和最厲害。這次要對付的就是大將軍，當你和大將軍對打時，千萬別動，當大將軍行近的時候，便向頭施以重擊，直至把大將軍打倒爲止。

當把大將軍打倒後，便立刻把操縱桿拉上右上角，此時，你應看到一個美女在面前，而你和她便會擁抱在一起。

筆者覺得這個遊戲聲、色、畫兼備，玩這個遊戲便如看電影一般，是八四年尾最後的佳作。■



SNOOPY II



軟件名稱：SNOOPY II
軟件類型：ACTION GAME
評　分：＊＊＊

大家一定記得筆者曾經介紹SNO—OPY 這個遊戲。今次再爲大家介紹SNOOPY II（SNOOPY續集）。

SNOOPY II和SNOOPY 並沒有多大分別，只是在開始時加上些有趣的漫畫，而今次的LEVEL 是由LEVEL 7至LEVEL 12，每個LEVEL 都比SN—OOPY更難玩，而每個LEVEL 更加上些時幻時眞的橋，令玩者提心吊胆。

在這裏，筆者不再費唇舌爲大家講解怎樣玩每一個LEVEL。只是教大家怎樣改遊戲中開始時的名字（例如：THIS DISK BELONGS TO LEE TAK WAH），讀者只要用COPY II PLUS 4.3C，然後用COPY DOS 功能，把一隻RUN HELLO的DOS COPY到SNOOPY II 中。然後用COPY II PLUS 的CHANGE BOOT PROGRAM ，將BOOT PROGRAM 轉爲STEX 便行，當BOOT 碟的時候，便能夠再寫另外的名字了。■

數恐龍學習加減運算

七歲以下兒童會感到有趣的遊戲

恐龍現在經已是絕種的動物。但大家再有機會在螢光幕上看見，而且還可以藉此來誘導幼童學習一些簡單的加、減數運算。

數恐龍是一個給兒童啓發思考的數學輔導程式，適宜七歲以下的兒童使用。

數恐龍程式是利用一些彩色的恐龍來組合成一些簡單的加、減數算術問題。

數恐龍的玩法有四種。程式執行後，按ESC 進入選擇表裏。0是選擇數目，1及2是加和減數，3是選擇加減混合運算。假如不按ESC ，程式會自動設定爲0模式。玩時，中途如按下ESC而又不作任何選擇，則程式會在幾秒鐘內返回0模式。

進行學習時，假如打入錯誤鍵，電腦會有聲發出。如按對了鍵，則會有一段恐龍音樂。

數恐龍包括了2個程式，一爲AP—PLESOFT主程式，另一爲繪劃恐龍圖形的機械碼程式。打入機械語言時需留意，圖形表有五種不同的恐龍圖形，（各位家長如懂圖形表造圖案原理，可自行設計各類圖形，　恐龍的圖形是19BIT濶，10BIT高，兩倍正常比例。）

各位家長留意一點，筆者在編寫這程序時並沒有加入記分的子程序，換言之，兒童在玩這遊戲時，毋用理會它的得失。筆者的目的是希望爲兒童編一個寓教育於娛樂的軟件。■

```
******************
**              **
**  DINO SHAPE  **
**              **
** A$4000,L$1E0 **
**              **
******************
4000-  07 00 10 00 51 00 71 00
4008-  9E 00 6A 01 C7 01 D6 01
4010-  08 18 08 18 08 18 08 18
4018-  25 0C 36 16 25 24 28 36
4020-  36 36 3E 57 09 08 38 24
4028-  24 24 15 36 36 35 0E 24
4030-  3C 2C 35 36 B6 9B 2D 25
4038-  24 24 24 35 36 36 2E 20
4040-  24 AC 36 2E 24 8C 36 25
4048-  64 36 25 24 08 30 36 36
4050-  00 20 0C 36 25 15 25 24
4058-  95 36 C5 21 24 08 36 36
4060-  25 2C 36 25 24 AC 32 76
4068-  24 C4 09 36 2E 35 2D 2D
4070-  00 18 08 24 08 36 36 25
4078-  2C 36 C5 21 C4 09 36 B6
4080-  25 24 24 C4 09 36 36 36
4088-  0E 20 64 36 2E 24 2C 36
4090-  2E 24 35 36 25 2C 2E 2C
4098-  2E 20 25 27 3F 00 08 18
40A0-  08 18 35 0E 24 0C 36 76
40A8-  24 24 2C 36 36 2E 24 24
40B0-  15 36 76 24 24 C4 21 0C
40B8-  36 36 36 2E 24 24 24 64
40C0-  36 B6 36 36 C5 21 24 C4
40C8-  09 18 C4 A9 B2 36 36 25
40D0-  24 24 35 36 36 C5 21 24
40D8-  15 36 C5 21 35 06 00 24
40E0-  00 08 08 08 48 09 2C 2E
40E8-  2C 2E 2C 2E 00 FF BF 24
40F0-  24 AC 32 25 05 00 FF 00
40F8-  08 08 08 08 48 29 2D 2D
4100-  3E 3F 3F 07 00 FF 00 08
4108-  08 08 08 48 31 0E 20 0C
4110-  30 36 36 0E 20 24 24 8C
4118-  12 76 20 04 00 08 08 08
4120-  48 09 2C 2E 2C 2E 2C 2E
4128-  00 FF BF 09 36 36 2E 24
4130-  24 AC 32 25 05 00 FF 00
4138-  08 08 08 08 48 29 2D 2D
4140-  3E 3F 3F 07 00 FF 00 08
4148-  08 08 08 48 31 0E 20 0C
4150-  30 36 36 0E 20 24 24 8C
4158-  12 76 20 04 00 08 08 08
4160-  48 09 2C 2E 2C 2E 2C 2E
4168-  00 FF 18 08 18 08 24 08
4170-  36 36 0E 24 08 18 60 96
4178-  32 96 21 20 C4 C1 C1 21
4180-  0C 16 96 36 76 20 24 C4
4188-  09 18 64 B2 32 36 0E 24
4190-  64 36 76 24 64 32 B6 0E
4198-  20 24 8C 36 0E 20 8C 0E
41A0-  24 18 08 60 32 36 76 08
41A8-  18 24 00 FF 00 2D 24 35
41B0-  2E 35 3F 36 27 3C 3F 00
41B8-  FF 00 2D 24 35 2E 2D 3E
41C0-  3F 36 27 3C 3F 04 00 2D
41C8-  24 2C 36 2E 2D 3E 3F 36
41D0-  3E 24 3C 3F 2C 00 2D 2D
41D8-  2D 35 3F 3F 3F 3F 2C 00
41E0-  F7
```

數恐龍學習加減運算

```
1   REM *******************
2   REM *    DINO MATH    *
3   REM *  BY COMPUHOUSE  *
4   REM *       &         *
5   REM *    MODIFY  BY   *
6   REM *                 *
7   REM *   LEE TAK WAH   *
8   REM *******************
10  FOR J = 1 TO 53: READ N: POKE
     767 + J,N: NEXT J: TEXT : HOME
     :SU = 0: FOR J = 1 TO 10:SU =
     SU +  PEEK (16383 + J): NEXT
     J: IF SU = 375 THEN 30
20  PRINT  CHR$ (4);"BLOAD DINO S
     HAPE"
30  POKE 233,64: POKE 232,0
35  GOSUB 345
40  GOSUB 60: GOSUB 170: GOSUB 23
     0: GOTO 40
50  IF F = 0 THEN 310
60  REM
70 A =  INT ( RND (1) * 7 + 1):AN
     = A: IF F = 0 THEN 160
80 B =  INT ( RND (1) * 7 + 1): IF
     F <  > 1 THEN 100
90 AN = A + B:S$ = "+": GOTO 150
100  IF F = 3 THEN 120
110 AN = A - B:S$ = "-": GOTO 150

120  IF  RND (1) > .5 THEN 140
130 AN = A - B:S$ = "-": GOTO 150

140 AN = A + B:S$ = "+"
150  IF AN < 1 OR AN > 9 THEN 60
160  RETURN
170  REM
180 SA = 40 + (7 - A) * 2:SB = 40
     + (7 - B) * 2:X1(1) = 140 -
     SA * (A / 2):X2(1) = 140 - S
     B * (B / 2): IF  RND (1) > .
     5 THEN X1(1) = X1(1) + 1:X2(
     1) = X2(1) + 1
190  HGR : HCOLOR= 3: SCALE= 2: ROT=
     0:SH =  INT ( RND (1) * 5 +
     1): FOR J = 1 TO A:X1(J) = X
     1(1) + (J - 1) * SA:Y1(J) =
     40: XDRAW SH AT X1(J),Y1(J):
     NEXT J: FOR J = 1 TO B:X2(J
     ) = X2(1) + (J - 1) * SB:Y2(
     J) = 120
200  XDRAW SH AT X2(J),Y2(J): NEXT
     J:SS = 7: IF S$ = "+" THEN S
     S = 6
210  XDRAW SS AT 131,71
220  HOME : VTAB 22: HTAB 15: PRINT
     A;S$;B;"=": RETURN
230  REM
240  VTAB 1: GET A$: PRINT "": IF
     A$ =  CHR$ (27) THEN  GOSUB
     340: RETURN
250  IF A$ = "E" THEN  TEXT : END

260 R =  VAL (A$): IF R = AN THEN
     280
270  GOSUB 290: GOTO 230
280  GOSUB 320: RETURN
290  REM
300  INVERSE : GOSUB 220: NORMAL
     : GOSUB 220
310  FOR I = 1 TO 4:P =  PEEK (49
     200) +  PEEK (49200) +  PEEK
     (49200) +  PEEK (49200): NEXT
     I: RETURN
320  REM
330  CALL 768: RETURN
340  REM
345  INVERSE : PRINT "DINO MATH";
     : NORMAL : PRINT  SPC( 18)"B
     Y COMPUHOUSE"
350 J = 0: TEXT : POKE 49168,1: VTAB
     10: PRINT  TAB( 16)"COUNT  0
     ": PRINT  TAB( 16)"ADD    1"
     : PRINT  TAB( 16)"SUBT   2":
     PRINT  TAB( 16)"MIX    3"
360 J = J + 1:P =  PEEK (49152): IF
     J > 150 THEN F = 0: RETURN
370  IF P > 175 AND P < 180 THEN
     390
380  GOTO 360
390 F = P - 176: POKE 49232,1: POKE
     49168,1: RETURN : DATA 169,1
     0,141,80,3,169,96,141,81,3,3
     2,174: DATA 239,165,159,201,
     5,16,6,24,105,2,76,15: DATA
     3,141,82,3,173,48,192,206,82
     ,3,238,82: DATA 3,206,82,3,2
     08,245,206,81,3,16,222,206
400  DATA 80,3,16,209,96
```

改動VTOC獲得102個SECTOR

莫信

各位讀者，你們可曾看過 COMPUTING AGE 第一期的「騰空更多磁碟空間——搬移 HELLO」一文及第七期的「十一條 SECTORS 」一文，將他們使用在一起，試過嗎？若你們將 HELLO HIDER和ELEVEN SECTORS 一起用的話，一些不能預知的錯誤便會發生了，因爲 HELLO 程式放在 TRACK $2的 SECTOR $5至 SECTOR $F ，而 ELEVEN SECTOR 則是將 TRACK $2 的 SECTOR $5 至 SECTOR $F解放出來，因而互相起了衝突，BOOT 碟時便有問題存在，爲了解決這個問題，最好便是將 HELLO 騰移至其他空間，考慮 TRACK $11 是存放 VTOC 及 CATALOG 的位置，而 VTOC 只佔用 SECTOR $0 ，其餘的都給安放 FILENAME用了，而每一個 SECTOR 可放7個 FILENAME，也就是說，一張磁碟最多可擁有7×15＝105 個 FILENAME，但我們往往只要 10至30 個 FILENAME，其餘的都被空了出來，這太浪費了，所以騰移 HELLO 至 TRACK $11 是最好的解決方法，首先各位讀者必須擁有像 LOCKSMI－TH，NIBBLE AWAY，BAG OF TRACK 或 CO－PY ］［ PLUS 等等能夠作磁碟內部的改動程式（筆者在此介紹 COPY ］［PLUS，而以下介紹的改動也是用 COPY ］［PLUS ，因它於這方面比較簡易）。

首先各位要有一片剛 INIT 的磁碟，這並不是硬性規定，只不過藉着它，我們了解比較容易，然後 RUN COP－Y］［PLUS（筆者手上所用的是 VER－4．0），哪一聲响了，看到MENU的選擇畫面，利用→鍵和←鍵選擇S－ECTOR EDITOR，按下RETURN，並選擇DISK A，隨之我們看到另一幅畫面，現在我們將那片剛INIT過的磁碟放於磁碟機內，並按P鍵及R鍵（P鍵代表PATC－HING，R鍵代表READING，畫面應有指示各鍵的用法，筆者不再在此詳述），電腦會詢問讀取那個TRACK及SECTOR，選定TRACK$11及SECTOR$0 ，我們會看到像圖一的畫面（畫面只可看到其中一部份，需要你們自己用I、J、K、M四個游標鍵來將其餘的畫面顯示出來）。

若看過「騰空更多磁碟空間——搬移HELLO」的朋友，相信亦會看過第一期的「DOS 3.3專欄之磁碟結構組織」，知道位址$01是存放索引TRACK的第一個SEC－TOR所屬的TRACK號數，而位址$02是索引TR－ACK的第一個SECTOR之SECTOR 編號，而位址$34是每磁碟上TRACK數目（正常情形下爲35（$23））。大約知道一些VTOC內部的位址後，再利用READ指令（按R鍵），看看第一個索引SECTOR（TRACK$11 SECTOR$F)將HELLO程式放在那兒，電腦將第一個索引SECTOR讀出後，我們會看到像圖二的畫面，留心觀看位址$0B及$0C，它指示我們HELLO程式是由T－RACK$12的SECTOR$F開始存放的，位址$0D是存放程式的檔案型號，而位址$0E至$12却是程式的名稱（對回ASCII碼，它便是HELLO），現我們再看看TRACK$12 SECTOR$F存放了甚麼（利用READ指令），可看到像圖三的畫面，由位址$0C開始，它分別存着程式第一個資料SECTOR的位置（TRACK$12 SEC－TOR$E），第二個資料SECTOR的位置（TRACK $12 SECTOR$D），……，第十一個資料SECTOR

圖（1）

```
00- 04 11 0F 03 00 00 FE 00
08- 00 00 00 00 00 00 00 00
10- 00 00 00 00 00 00 00 00
18- 00 00 00 00 00 00 00 00
20- 00 00 00 00 00 00 00 7A
28- 00 00 00 00 00 00 00 00
30- 12 01 00 00 23 10 00 01
38- 00 00 00 00 00 00 00 00
40- 00 00 00 00 FF FF 00 00
48- FF FF 00 00 FF FF 00 00
50- FF FF 00 00 FF FF 00 00
58- FF FF 00 00 FF FF 00 00
60- FF FF 00 00 FF FF 00 00
68- FF FF 00 00 FF FF 00 00
70- FF FF 00 00 FF FF 00 00
78- FF FF 00 00 00 00 00 00
80- 00 1F 00 00 FF FF 00 00
88- FF FF 00 00 FF FF 00 00
90- FF FF 00 00 FF FF 00 00
98- FF FF 00 00 FF FF 00 00
A0- FF FF 00 00 FF FF 00 00
A8- FF FF 00 00 FF FF 00 00
B0- FF FF 00 00 FF FF 00 00
B8- FF FF 00 00 FF FF 00 00
C0- FF FF 00 00 00 00 00 00
C8- 00 00 00 00 00 00 00 00
D0- 00 00 00 00 00 00 00 00
D8- 00 00 00 00 00 00 00 00
E0- 00 00 00 00 00 00 00 00
E8- 00 00 00 00 00 00 00 00
F0- 00 00 00 00 00 00 00 00
F8- 00 00 00 00 00 00 00 00
```

改動 VTOC 獲得 102 個 SECTOR

圖（2）

00-	00	11	0E	00	00	00	00	00
08-	00	00	00	12	0F	02	C8	C5
10-	CC	CC	CF	A0	A0	A0	A0	A0
18-	A0	A0	A0	A0	A0	A0	A0	A0
20-	A0	A0	A0	A0	A0	A0	A0	A0
28-	A0	A0	A0	A0	09	00	00	00
30-	00	00	00	00	00	00	00	00

圖（3）

00-	00	00	00	00	00	00	00	00
08-	00	00	00	00	12	0E	12	0D
10-	12	0C	12	0B	12	0A	12	09
18-	12	08	12	07	12	06	12	05
20-	00	00	00	00	00	00	00	00
28-	00	00	00	00	00	00	00	00

的位置（TRACK＄12 SECTOR＄5），隨後的都是＄00，表示再沒有資料SECTOR了，現在我們可以看到，一個佔11個SECTOR的HELLO程式，當INIT時，會被安放在TRACK＄12的SECTOR＄5至SECTOR＄F，而它的排列次序是由SECTOR＄F開始，然後SECTOR＄E，最後是SECTOR＄5（當然，每人的HELLO程式的長度都有不同，若它只有9個SECTOR，則它只佔TRACK＄12的SECTOR＄7至SECTOR＄F，如此類推），到現在爲止，我們還未對HELLO搬移，到底我們將HELLO程式移至那兒呢？很簡單，將HELLO移至TRACK＄11的SECTOR＄5至SECTOR＄F便成了，換句話說，我們是要將TRACK＄12 SECTOR＄F 移至TRACK＄11 SECTOR＄F，將TRACK＄12 SECTOR＄E移至TRACK＄11 SECTOR＄E，如此下去，直至將所有資料SECTOR全搬到TRACK＄11去，但未作這個搬移前，我們先要將TRACK＄11SECTOR＄F移至同軌道的SECTOR＄4，爲什麼要這樣？若我們不先移走SECTOR＄F，當搬移HELLO時會被掩蓋，消失掉了，再參看圖二，位址＄01及＄02將下一個索引SECTOR指向TRACK＄11SECTOR＄E，而SECTOR＄E已被搬移的HELLO佔用了，所以必須將它改掉，而位址＄0B，＄0C是存着HELLO程式的開始SECTOR，搬移後，開始SECTOR已改，因此也要將它改動，首先利用游標鍵移至位址＄02，按H鍵，再按03，將索引POINTER改爲＄03，然後按RETURN鍵，表示改動完畢，再將游標移至＄0B，按H鍵，寫入＄11，再按RETURN，表示第二次改動完畢，但此時磁碟內並未接受這個改動，所以要按W鍵（表示WRITING），選擇TRACK＄11 SECTOR＄04，選擇好後，按RETURN，改動便眞正完成了，可以開始移動HELLO了，我們用READ指令將TRACK＄12 SECTOR＄5讀出，然後再WRITE指令讀到TRACK＄11 SECTOR＄5去，重覆這個步驟，直至將SECTOR＄5至SECTOR＄E全部搬移，SECTOR＄F呢？參看圖三，SECTOR＄F是存放資料SECTOR的SECTOR，所以SECTOR＄F須先改動才作搬移，其實很容易，只要將＄12全改爲＄11便成了，利用H指示改動它，再WRITE到TRACK＄11 SECTOR＄F去，搬移HELLO便算告一段落，第一個索引SECTOR亦已被搬到SECTOR＄4去，我們還有沒有什麼未做的嗎？看看題目，「改動VTOC獲得102 SECTOR」，我們還未改動VTOC啊，用READ指令，再看看TRACK＄11SECTOR＄0，參看圖一，位址＄01及＄02，它指向第一個索引SECTOR的位置是TRACK＄11 SECTOR＄F，今第一個索引SECTOR已被改到SECTOR＄4去，所以位址＄02便需要改作＄04，但題目寫明獲得102個SECTORS，現今只多了11SECTORS，其餘的91個SECTORS呢？只要我們將位址＄34由＄23改作＄28，位址＄40和＄41改作＄FF，＄E0，位址＄80和＄81改作＄FF、＄FF，最後將位址＄C4至＄D5順序改成＄FF、＄FF、＄00、＄00、＄FF、＄FF、＄00、＄00、＄FF、＄FF、＄00、＄00、＄FF、＄FF、＄00、＄00、＄FF、＄FF，便大功告成了。

位址＄80、＄81是指示TRACK＄12的空SECTOR位址對映，而TRACK＄12的HELLO程式已被搬移，自然TRACK＄12便沒有用處，所以將它改成＄FF、＄FF，而改動位址＄40、＄41爲＄FF、＄E0，效果便如用ELEVEN SECTOR 程式一樣，將TRACK＄2的SECTOR＄5至SECTOR＄F解放出來，這樣我們便多了十一個SECTORS了，最後改動位址＄34爲＄28，是要宣告將只有35TRACKS的DOS3.3改爲40TRACKS，而改動位址＄C4至＄D5便是要配合這個改變，解放5條TRACK，總共有5×16＝80個SECTORS，加上將HELLO搬移和TRACK＄2的11SECTORS，便總共增加了閣下DOS3.3磁碟102個SECTORS了。（最後，筆者警告各位，若你所擁有磁碟機並不能行40TRACK的磁碟片的話，則改動位址＄34和位址＄C4至＄D5便不可以，若要磁碟機強行40TRACK，會對你的磁碟片或磁碟機等造成傷害。）■

後記：各位在改動VTOC時，必須要將TRACK 2 SECTOR 2位址的值＄BD改爲＄04，否則BOOT DOS時會有問題。

一本蘋果電腦用家必備雜誌

每期內容包括：

- 多個價值無可估計的軟件程式
- 專文深入討論週邊硬件的用途
- 最新的軟件介紹及評論
- 多個有深度而題材廣泛的專欄
- 還有其他精彩內容未能在此一一盡錄



第一期　COMPUTEXT 文字處理系統／電腦時代個人月曆／搬移 HELLO 騰空更多磁碟空間／重現被刪除之檔案／幼童數學四則運算練習／如何選購 AP II 電腦／上、下、左、右移動文字畫面／磁碟的結構／時日相距計算程式／過四關／增加英文字彙能力的遊戲／眞假隨機數………

第二期　彩色電視監視器色彩測試／沙蟹幻方局／貨櫃處理／美術字創造器／磁碟修護工具／數學套件程式／程式轉載器／如何用 FID 抄錄有保護之磁碟／AP II 擴展插座之功能及應用探討／BAG OF TRICK 的拷貝辦法遊戲積分紀錄／抄錄程式的好幫手 WILDCARD ………



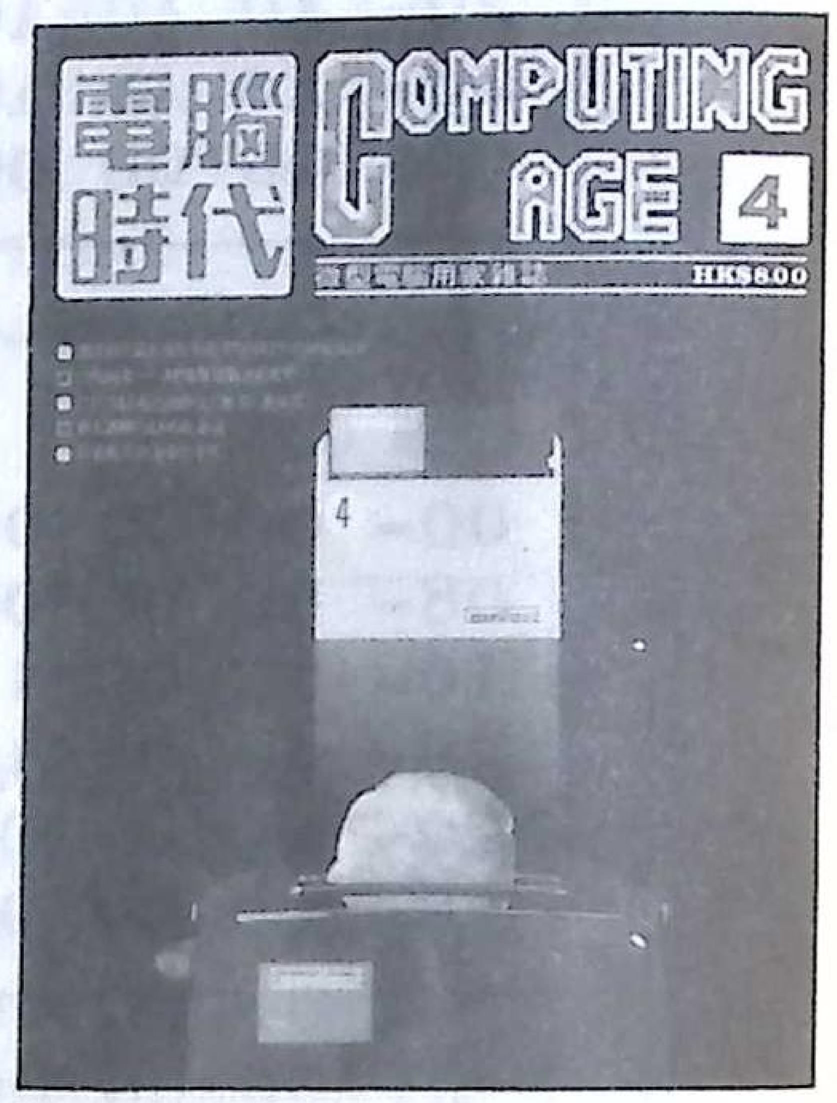
第三期　神經鬆弛體操／磁碟索引表封套製造器／雜誌目錄分類歸檔系統／DOS BOOT 碟的原理／動量不滅定律／幼童認數字初階／步步追踪／門外漢選購印字機／音樂產生的原理／蘋果編曲器／DOS 殺手／DOS 自動同步追踪器／圍棋遊戲／機械語言程式輸入方法／賽跑遊戲………

第四期　將 RAM 咭作爲磁碟機使用／太陽系行星移動探測系統／如何修補 DOS／立體迷宮／穿梭機特別任務／應用數學套件（二）／AP II 微電腦進化史／學前兒童幼稚園／觀看 TEXT FILE／DOS WRITER／電腦動畫製作原理／英文串字練習／PRODOS 操作系統介紹（上）／黑房計時器／失落的 STEP 及 TRACE ………



第五期　私人約會備忘日曆／立體空間動畫顯像系統／將 CP／M 檔案轉爲 DOS 檔案／80字行咭縱橫談／PRODOS 操作系統指令介紹（下）／6502 十三種存取資料地址模式／幼童數學練習數字咭／加數機大登場／神偷智破夾萬／隔山打牛／太空船降落任務／資料庫資料記錄原理／濃縮圖片儲存空間／蘋果萬花筒／DOS 指令簡化器／CP／M 專欄………

第六期　妙用無窮用的128K RAM咭／APPLESOFT 擴展版本／私人會計系統／APPLESOFT 程式語言偵錯工具／DOS 緩衝器剖析／GO II／迷宮搜秘／太空侵略戰／亡命飛車／累加遊戲／神奇數學方塊／砌積木／機械人全面認識／CP／M 及 PASCAL 專欄／任意定義鍵盤／布倫時鐘／完整 CATALOG／爲磁碟命名………



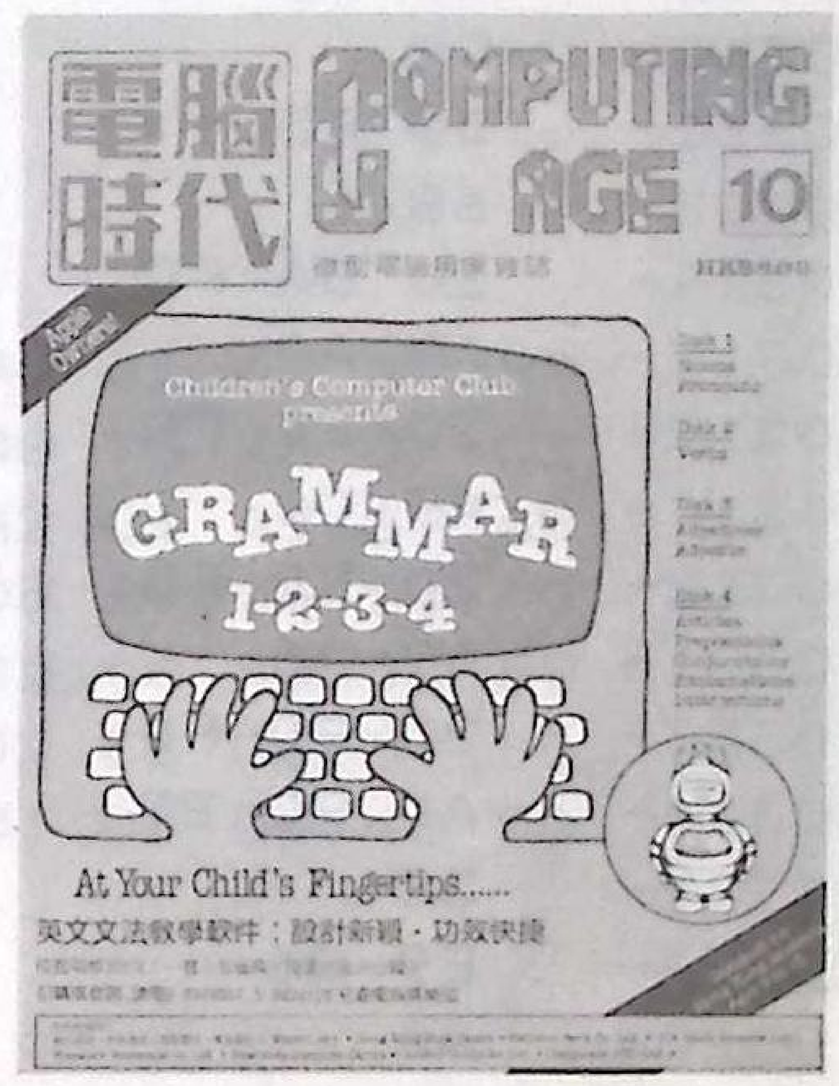




訂閱收費表（包括郵費在內）

	本港	澳門、中國、東南亞區	其他地區	
雜誌加程式磁碟（十二期）	HK$455 ——	HK$500或US$65 HK$630或US$80	HK$530或US$68 HK$675或US$87	平郵 空郵
雜誌（十二期）	HK$96 ——	HK$118或US$15 HK$218或US$28	HK$140或US$18 HK$250或US$32	平郵 空郵





第七期 會說中文的蘋果／私人財政預算系統／私人財政收入系統／ PRODOS專欄／再談 80 字行咭／高解像圖轉載器／螢幕文本編輯處理器／蘋果杯大賽車（用 GRAPHICS MAGICIAN 編寫的遊戲）／高爾夫球遊戲／十一個 SECTOR／空戰驚魂／高解像字符繪劃器／ DOS 3.3專欄⋯

第八期 APPLESOFT滙編器／編印單個 LABEL／Z80 咭的認識／將 INTEGER 轉爲 APPLESOFT 方法／ RESET 鍵特殊問題／高解像頁處理工具箱／眞正要動腦筋的小學四則運算／七海霸王（麥法基專欄）／原子黑洞／蛇之噩夢／有廿四種功能的 DOUBLE TAKE／自動三角函數曲綫繪畫系統／兩欄索引顯示⋯⋯⋯





第九期 磁碟檔案黃頁系統／增強鍵盤功能的 COMPUSOFT ／混合機械語言 APPLESOFT 程式的技術初探／再談 Z 80 咭地址配對方法／耐人研究的變數(上)／奇妙的&專欄／ APPLE TOOL KIT 各項優異功能／古堡之謎／七海霸王（下）／高空擲物／新 DOS 命令改換程式／捉迷藏／有趣的排序⋯⋯⋯

第十期 可隨意訂定格式之資料紀錄系統／ 16 K RAM 咭潛能發揮的竅門技巧／跳越加速器／ MACRO 鍵盤／機械語言入門／ 40磁軌工具程式之（一）／速度奇快的 TURBO PASCAL／ LISA 2・5 細談／將 DOS 移上 RAM 咭／培養兒童記憶能力／初中代數式運算／萬用數基轉換器／太空戰士（麥法基專欄）／射鴨仔／ 16 欄機械碼編印⋯⋯





第11期 加速蘋果運作速度的多種方法探討／將 WORDSTAR 放入 RAMDIST使用（128 K RAM 咭應用篇）／魔域迷宮 ／VISICALC 專輯／家庭購物備忘及食物用品存貨紀錄系統／ PRINT SHOP 評介／電子辭典典範 SENSIBLE SPELLER／LOCK IT UP使用方法／ JACKIES HEAD ／抄碟拆碟資料館／美化程式列表編印／英文語句文法分析練習系統／巨蟹求生記⋯

第12期 RAM 咭控制中心／家庭開支及個人消費管照系統／磁頭清潔器／諾曼第登陸戰／聲音和畫面技巧(麥法基專欄)／圖型建造師／符號逆輸計數機／電子郵箱／ LOCKSMITH 使用方法／ CP／M 55 個公用程式 POWER ／教育軟件評介專欄／ DTEST——測試你對 DBASE Ⅱ 的認識／加強 WORDSTAR 編印字體模式效果／有多項功能的 AMPERSAND 程式／工兵掘地雷／ E－Z DRAW 評介⋯

更正

第13期〔保護程式技巧之一——搬移V TOC法〕P.76第二段〔⑴至此……即選定TRACK 1 SECTOR B(第八行)……〕應改爲SECTOR 9。

同期第15頁〔發聲的鍵盤〕線路圖中的C3電容值漏印應是0.47μ。

第十三期PRODOS縱橫的PREFIX 命令中漏登一個列表；現補載如下：

```
0300-   20 41 03    JSR   $0341
0303-   90 30       BCC   $0335
0305-   48          PHA
0306-   A2 00       LDX   #$00
0308-   BD 36 03    LDA   $0336,X
030B-   F0 06       BEQ   $0313
030D-   20 ED FD    JSR   $FDED
0310-   E8          INX
0311-   D0 F5       BNE   $0308
0313-   68          PLA
0314-   20 DA FD    JSR   $FDDA
0317-   A9 A0       LDA   #$A0
0319-   20 ED FD    JSR   $FDED
031C-   A9 C0       LDA   #$C0
031E-   20 ED FD    JSR   $FDED
0321-   A9 A0       LDA   #$A0
0323-   20 ED FD    JSR   $FDED
0326-   AD 9C BF    LDA   $BF9C
0329-   38          SEC
032A-   E9 06       SBC   #$06
032C-   AA          TAX
032D-   AD 9D BF    LDA   $BF9D
0330-   E9 00       SBC   #$00
0332-   20 41 F9    JSR   $F941
0335-   60          RTS
0336-   CD CC C9    CMP   $C9CC
0339-   A0 C5       LDY   #$C5
033B-   D2          ???
033C-   D2          ???
033D-   CF          ???
033E-   D2          ???
033F-   A0 00       LDY   #$00
0341-   AD 9A BF    LDA   $BF9A
0344-   D0 2E       BNE   $0374
0346-   AD 30 BF    LDA   $BF30
0349-   8D 77 03    STA   $0377
034C-   20 00 BF    JSR   $BF00
034F-   C5 76       CMP   $76
0351-   03          ???
0352-   90 02       BCC   $0356
0354-   38          SEC
0355-   60          RTS
0356-   AD 7B 03    LDA   $037B
0359-   29 0F       AND   #$0F
035B-   D0 05       BNE   $0362
035D-   AD 7C 03    LDA   $037C
0360-   D0 F2       BNE   $0354
0362-   AA          TAX
0363-   E8          INX
0364-   8E 7A 03    STX   $037A
0367-   A9 2F       LDA   #$2F
0369-   8D 7B 03    STA   $037B
036C-   20 00 BF    JSR   $BF00
036F-   C6 8C       DEC   $8C
0371-   03          ???
0372-   B0 E0       BCS   $0354
0374-   18          CLC
0375-   60          RTS
0376-   02          ???
0377-   00          BRK
0378-   7B          ???
0379-   03          ???
037A-   00          BRK
037B-   00          BRK
037C-   00          BRK
037D-   00          BRK
037E-   00          BRK
037F-   00          BRK
0380-   00          BRK
0381-   00          BRK
0382-   00          BRK
0383-   00          BRK
0384-   00          BRK
0385-   00          BRK
0386-   00          BRK
0387-   00          BRK
0388-   00          BRK
0389-   00          BRK
038A-   00          BRK
038B-   00          BRK
038C-   01 7A       ORA   ($7A,X)
038E-   03          ???
038F-   14          ???
0390-   15 16       ORA   $16,X
0392-   17          ???
0393-   18          CLC
0394-   19 1A FF    ORA   $FF1A,Y
0397-   FF          ???
0398-   FF          ???
0399-   FF          ???
039A-   FF          ???
039B-   FF          ???
039C-   FF          ???
039D-   FF          ???
039E-   FF          ???
039F-   FF          ???
03A0-   FF          ???
03A1-   1B          ???
03A2-   FF          ???
03A3-   1C          ???
```

將程式隱藏在記憶體裡

相信大多數的朋友，對BASIC 語言並不會感到陌生，擁有電腦的人仕，都知道它易學易用，大受用家歡迎，但BASIC 語言却極容易被人拆解、寫程式寫得這麼辛苦，一下子便被人抄去了，氣死人。

HIDE是特別爲BASIC 語言而設計的程式，它的特點是當你由磁碟或磁帶LOAD 程式入電腦後，用LIST 命令觀看程式內寫了什麽東西時，它祇顯示第一行程式，其餘的會受到破壞，像被DELETE 一樣，這時再用RUN命令執行程式時，GAME OVER 。而且連續二次用LIST命令，就好像用NE—W命令一樣，第一行程式也玩完了。當然LOAD 程式入電腦後，立即用—RUN命令執行，程式並不會受到破壞。

用這個HIDE 程式，有一點是需要注意的，你們所寫的第一行程式一定要用REM命令，而且隨REM命令之後，最少要有十個空格（只需按十次SP—ACE BAR 便成了），因爲HIDE 程式會將一些資料碼塡入第一行程式。最後，你們必須於寫完你的程式後，才可用＄9000G執行HIDE 程式，還有的是程式執行完後，切不可用LIST 命令，用SAVE 將你的程式載入磁帶或磁碟。

但這個HIDE 程式也有它的缺點，當用LIST命令只要避開第一行來執行；一樣的可以觀看到程式到底是寫了些什麽東西。以小玩意來看，它是極具趣味性的。

本程式在程式磁碟中的FILENAME是HIDE。■

```
***************
* 1985.JAN.5 *
*     HIDE    *
***************
9000-  A2 00 A0 00 A1 67 A8 88
9008-  88 88 88 88 98 85 07 C1
9010-  67 90 0C A2 01 A1 67 A8
9018-  88 98 85 08 4C 25 90 A0
9020-  01 B1 67 85 08 A0 00 B9
9028-  34 90 91 07 C8 C0 04 D0
9030-  F6 60 00 01 0D 04 46 50
9038-  A2 90
*
```

捉蟲訓練— DOS指令的執行

這個捉虫訓練問題是在APPLE—SOFT BASIC裏執行DOS指令，在編寫這類程序時，時常會遇到這類的問題，但知道原因的人並不多。

這個程序首先會要求使用者輸入檔案名稱，然後建立一個文字檔，內有一百個〝TEST〞字樣，最後在螢幕上打出〝THIS IS FINISHED.〞訊息。但是以下的程序却不能完整地運行，究竟「虫」在那裏呢？

我們歡迎讀者提供〔捉蟲訓練〕問題。程式長度沒有限制，以越簡短越好。程式必須要以APPLESOFT 寫成，並符合下列各項要求：

1. 程式必須是APPLESOFT ，任何POKE或CALL機械語言的敍述必須正確。
2. 程式必須能出現某些結果，而不可出現SYNTAX ERROR。
3. 所有問題必須附加簡短說明及答案。

假如閣下的作品獲得刊登，我們將會送出十二期免費雜誌給你或將你的訂閱延長一年。

所有投稿請寄九龍中央郵政信箱C. P. O BOX 71193 。信封面請寫上〔投稿捉蟲〕。■

```
10 D$ =  CHR$ (4): REM  CTRL-D
20  INPUT "FILENAME"FI$
30  PRINT D$"NOMON C,I,O"
40  PRINT D$"OPEN"FI$
50  PRINT D$"WRITE"FI$
60  FOR I = 1 TO 100
70  PRINT "TEST ";
80  NEXT I
90  PRINT D$"CLOSE"
100  PRINT "THIS IS FINISHED."
110  END
```

答案：只須在第八十行的NEXT I 之後加上一個PRINT，即80 NEXT I ：PRINT因爲CTRL—D一定要是CARRIAGE RETURN之後的第一個字元才會被視爲DOS COMMAND。所以以上的程序的文字檔根本沒有CLOSE 上，以後的訊息便PRINT 入檔案中了。

徵求一行程式有獎遊戲

在上月我們收到許多〔一行程式〕的投稿，其中不乏精彩的作品。

這個遊戲旨在鼓勵大家動腦筋，利用最簡短的程式，設計一些有創意的效果。

今次未刊登你的作品的讀者請勿灰心，可能我們會在下一期登出。

各位讀者寫來程式時，必須用多行的形式寫來（參考本文的各位得獎例子），而且要書寫清楚。投稿請寄來CPO BOX 71193 。入選者可獲得任何一期的程式磁碟一張（請註明要的磁碟）。來稿信封請勿封口。■

本程式能在高解像畫面繪出一尾在暢泳的魚兒。鍾達維讀友可取得第十三期磁碟乙張。

```
30    HGR2
  : FOR I = 0 TO 360
  : X = I / 1.4 + 20
  : Y = SIN (I) * 10 + 100
  : HCOLOR= 7
  : HPLOT X,Y TO X - 6,Y + 6 TO X
     - 18,Y - 6 TO X - 18,Y + 6
     TO X - 6,Y - 6 TO X,Y
  : HPLOT X - 4,Y
  : FOR J = 0 TO 100
  : NEXT
  : HCOLOR= 0
  : HPLOT X,Y TO X - 6,Y + 6 TO X
     - 18,Y - 6 TO X - 18,Y + 6
     TO X - 6,Y - 6 TO X,Y
  : HPLOT X - 4,Y
  : NEXT
  : RUN
```

此爲一繪圖工具程式，簡單而實用。D是繪畫模式，X爲擦出模式，I，J，K，M 用來控制上、下、左、右移動。轉換模式時先按X或D。杜文輝君可獲得我們的第七期程式磁碟。

```
5     HGR2
  : X = 140
  : Y = 96
  : H = 3
  : FOR I = 1 TO 2 STEP 0
  : GET A$
  : H = H - 3 * (A$ = "X" AND H = 3)
     + 3 * (A$ = "D" AND H = 0)
  : HCOLOR= H
  : X = X - (A$ = "J" AND X > 0) + (A
     $ = "K" AND X < 279)
  : Y = Y - (A$ = "I" AND Y > 0) + (A
     $ = "M" AND Y < 191)
  : HPLOT X,Y
  : NEXT
```

本程式是個吃角子老虎機，玩者需鍵入落注注碼，電腦會隨機顯示出三個英文字母（由A—F），代表ＪACKPOT的顯示，跟著便顯示結果。

陳文緯讀者可取得第十一期程式磁碟。

```
10    HOME
  : INPUT "BET=$";K
  : FOR I = 1 TO 3
  : A(I) = INT ( RND (1) * 6) + 1
  : PRINT CHR$ (A(I) + 64);"    ";
  : NEXT
  : PRINT
  : VTAB 3
  : PRINT "YOU LOSE$"K / 2
  : IF A(1) = A(2) OR A(2) = A(3)
     OR A(1) = A(3) THEN VTAB 3
  : PRINT "YOU GAIN "K / 2
  : IF A(1) = A(2) AND A(2) = A(3)
     THEN PRINT "YOU GAIN"K"    "
```

以下是一個〔計時器〕程式，用作計時之用。當時間一到，便會有聲音提示你。執行程式時，電腦會詢問响鬧的時間。舉例，如果希望〔計時器〕在一時十五分後响鬧，輸入75即可。跟著再按下任何鍵，便開始計時。HOWARD HO讀者可贏取得一張第十三期的磁碟。

```
10    HOME
  : INPUT "TIME";M
  : CALL - 756
  : HOME
  : I = - 16336
  : POKE 34,11
  : POKE 35,11
  : FOR K = 0 TO M - 1
  : HOME
  : HTAB 16
  : PRINT K; CHR$ (255);
  : POKE 32,17
  : FOR R = 0 TO 59
  : X = PEEK (I) - PEEK (I)
  : PRINT R;
  : FOR H = 1 TO 739
  : NEXT H
  : PRINT
  : NEXT R
  : POKE 32,0
  : NEXT K
  : HOME
  : FOR K = 1 TO 200
  : HTAB 16
  : PRINT "TIMEUP"
  : S = PEEK (I)
  : NEXT K
```

各期程式磁碟磁帶總目錄

1
1 突破傳統的鍵盤
2 捉球遊戲
3 勇闖電腦室
4 迷宮救美
5 將 APPLESOFT 變爲 FORTRAN
6 上、下、左、右移動文字畫面
7 測試工具— BOOLEAN 程式
8 騰空更多磁碟空間一搬移HELLO
9 重現刪除之檔案
10 電腦時代月曆
11 COMPUTEXT 文字處理
12 時日相距計算程式
13 增強英文字滙能力
14 幼童數學四則運算練習
15 過四關

2
1 貨櫃處理
2 沙蟹幻方局
3 巧破密令
4 美術字繪劃器
5 小階字母處理
6 程式轉載器
7 彩色電視／監視器色彩測試
8 保留遊戲積分記錄
9 數學套件程式
10 磁碟修護工具
11 加送電腦時代月曆圖像一個

3
1 DOS 的自動同步追踪器
2 動量不滅定律模擬法
3 自製磁碟索引表封套
4 蘋果編曲器
5 幼童認數字初階
6 賽跑遊戲
7 步步追踪
8 DOS KILLER
9 圍棋遊戲
10 巴哈的古典樂章名作
11 旋轉螢幕畫面
12 雜誌目錄分類歸檔系統
13 神經病醫生一神經鬆弛體操

4
(1) DOS 操作指令簡化器
(2) TEXT FILE 有些甚麼內容?
(3) 立體迷宮
(4) 英文串字練習
(5) 如何以機械語言來修補 DOS
(6) 電腦動畫製作原理。
(7) 豐富設計的學前兒童教育程式
(8) 應用數學套件程式(二)
(9) DOS WRITER
(10) 穿梭機哥倫比亞號機
(11) 找回失落了的 STEP 及 TRACE
(12) TRACER控制常式
(13) 將 RAM 卡作爲磁碟機使用
(14) 黑房計時器
(15) 太陽系模擬系統

9
(1) 磁碟檔案黃頁系統。
(2) 增強鍵盤輸入及螢幕輸出功能的系統
(3) 將工具程式按放入 DOS 佔用的三條磁軌上
(4) 快速尋找及磁碟矩陣儲存與提取
(5) 新 DOS 命令改換程式
(6) 任意設定目錄索引顯示模式
(7) 捉迷藏
(8) 古僊之謎
(9) 高空瓣物
(10) 有趣的排序
(11) 捉蟲問題及答案
(12) 第八期捉蟲問題答案
(13) 印出高解像數目字

5
1 私人約會備忘日曆
2 三度空間顯像系統
3 將 CP／M 檔案轉爲 DOS
4 蘋果萬花筒
5 高解像度文字產生器
6 修訂 DOS 操作指令簡化器
7 濃縮圖片儲存空間
8 幼童數學練習—— 數字咭
9 加數機大登場
10 備忘 MEMO紙
11 光速計數器
12 剩餘磁碟空 SECTOR數目
13 16進制輸入數字鍵盤
14 問題棋盤
15 神偷智破夾萬
16 隔山打牛
17 大空船降落任務

6
(1) 私人會計系統。
(2) APPLESOFT 擴展版本。
(3) GO } [HELLO 的程式
(4) APPLESOFT 偵錯系統。
(5) DOS 指令再修訂版本。
(6) 布倫時鐘。
(7) AMPERSAND 數基轉換器
(8) 完整索引鍵。
(9) 砌積木。
(10) 神奇數字方塊。
(11) 亡命飛車。
(12) 迷宮搜秘。
(13) 太空怪侵略戰。
(14) 累加遊戲。
(15) 爲磁碟命名。
(16) 任意定義鍵盤。
(17) 捉蟲訓練問題。

7
(1) 會說話的蘋果
(2) 私人財政預算系統
(3) 私人收入會計系統
(4) 高解像字符繪劃器
(5) 圖像轉載器
(6) 螢幕文本編輯處理器
(7) 濃縮圖片程式修訂加強版本
(8) 蘋果杯大賽車
(9) 十六個難度的食鬼遊戲
(10) 高爾夫球遊戲。
(11) 空戰驚魂
(12) 美麗圖案設計家
(13) 更多色彩
(14) 列印的困擾
(15) 十一條空 SECTOR
(16) 捉蟲訓練
(17) 80字行咭例子程式

8
(一) APPLESOFT滙編器
(二) 最簡單的拆帶程式
(三) 英文打字練習
(四) 編印單個LABEL
(五) 高解像頁處理工具箱
(六) 簡易快速的記憶移動方法
(七) 眞正要動腦筋的小學四則運算
(八) 七海霸王 (DEPTH CHARGE)
(九) 原子黑洞
(十) 蛇之噩夢
(十一) 收銀機
(十二) 改良的閃動訊息
(十三) 兩欄索引顯示
(十四) 自動三角函數曲綫繪劃系統
(十五) 任意鍵盤 (三個)
(十六) 每期捉蟲問題
(十七) 奇妙的 &

各期程式磁帶

編號：C#1
1、捉球遊戲
2、勇闖電腦室
3、迷宮救美
4、過四關
5、增強英文字滙能力
6、幼童數學四則運算
7、時日相距計算程式

編號：C#2
1、沙蟹幻方局
2、貨櫃處理
3、數學套件程式
4、巧破密令
5、彩色電視／監視器色彩測試

編號：C#3
1、賽跑遊戲
2、步步追踪
3、圍棋遊戲
4、動量不變定律模擬方法
5、幼童認數字初階

編號：C#4
1、豐富設計的學前兒童教育程式
2、應用數學套件程式之二
3、立體迷宮
4、穿梭機哥倫比亞號
5、英文串字練習

編號：C#5
1、加數機大登場
2、神偷智破夾萬
3、隔山打牛
4、太空船降落任務
5、蘋果萬花筒
6、幼童數學練習——數字咭

編號：C#6
1、累加遊戲
2、亡命飛車
3、迷宮搜秘
4、太空侵略戰
5、切積木
6、神奇數學方塊

編號：C#7
(一) 十六種難度的食鬼遊戲
(二) 更多色彩
(三) 空戰驚魂
(四) 捉虫問題
(五) 捉虫答案

編號：C#8
(一) 最簡單的拆帶程式
(二) 英文打字練習
(三) 原子黑洞
(四) 蛇之噩夢
(五) 收銀機
(六) 任意鍵盤

精選程式磁碟

編號：CA00A
內容：遊戲及教育程式
遊戲程式
1、捉球遊戲（第一期）
2、勇闖電腦室（第一期）
3、迷宮救美（第一期）
4、過四關（第一期）
5、沙蟹幻方局（第二期）
6、貨櫃處理（第二期）
7、賽跑遊戲（第三期）
8、步步追踪（第三期）
9、圍棋遊戲（第三期）
10、立體迷宮（第四期）
11、穿梭機哥倫比亞號（第四期）
教育程式
12、增強英文匯能力（第一期）
13、幼童數學四則運算（第一期）
14、數學套件程式（第二期）
15、動量不滅定律模擬法（第三期）
16、幼童認數字初階（第三期）
17、豐富設計的學前兒童教育程式（第四期）
18、應用數學套件程式之二（第四期）

編號：CA00B
內容：應用程式精選
1、電腦時代月曆（第一期）
2、COMPUTEXT 文字處理（第一期）
3、美術學繪劃器（第二期）
4、彩色電視／監視器色彩測試（第二期）
5、保留遊戲積分紀錄（第三期）
6、自製磁碟索引封套（第三期）
7、APPLE 編曲器（第三期）
8、雜誌目錄分類歸檔系統（第三期）
9、DOS WRITER（第四期）
10、電腦動畫製作原理（第四期）
11、將 RAM 咭作爲磁碟機使用（第四期）

歡迎補購各期
程式磁碟磁帶或雜誌

補購請到——
上海街438號同珍商業中心11字樓1104室（上海街／登打士街交界）
電話：3-312007

長期訂戶啟事

由於今期刊出的多篇文章程式是CP／M版本，其磁碟版本需要有不同的格式FORMAT，故此我們出版有2張程式磁碟，兩張都是雙面灌錄。磁碟編號分爲#14A及#14B。有關內容可見磁碟目錄。

雜誌及程式磁碟的長期訂戶將只收到#14A磁碟，不過，只要你們寄來一張空磁碟（要雙面有效），及一個貼有一元面額郵票的回郵信封，我們便會將#14B寄到府上，無須額外付費。記住，空磁碟請寄往九龍中央郵政信箱CPO BOX 71193號，信封面請註明〔長期訂戶〕及〔訂閱戶號碼〕，我們會快速辦理。又或者可前來讀者服務部攞取。

如讀者在服務部同時購買第14期兩張磁碟則可獲折扣優待，每張只須付出25元。購買一張則沒有。此折扣優待至二月底止。■

各期程式磁碟磁帶總目錄

第11期程式磁碟內容：

(1) 家庭購物備忘及存貨紀錄系統
(2) 可隨意訂定之資料紀錄系統（下）
(3) 奇妙的＆專欄——矩陣定値
(4) 40軌磁碟工具程式之(二) 單軌 INIT
(5) 兩個自動減省程式佔用之記憶空間應用程式
(6) 對我說
(7) 趣意盎盎的歡樂棒
(8) 英文語句文法分析練習系統
(9) 一行程式示例（共七個）
(10) 美化程式列表
(11) JACKIES HEAD
(12) 魔域迷宮
(13) 巨蟹求生記
(14) 自動日子記錄
(15) 空SEOCTOR圖表
(16) 捉虫訓練——FACTORIAL！

第10期程式磁碟內容：

(1) 可隨意訂定格式之資料紀錄系統
(2) 太空戰士
(3) MARCO鍵盤
(4) 射鴨仔
(5) 奇妙的＆——快速分類整理程式
(6) 40軌磁碟工具程式設計
(7) 變數表顯示
(8) 跳越加速器
(9) IQ方塊
(10) 快速檔案搜索
(11) 將DOS移上RAM咭
(12) 培養兒童的記憶能力
(13) 初中代數式運算
(14) 萬用數基轉換器
(15) 十六欄機械碼印
(16) 捉蟲問題及答案

第13期程式磁碟內容：

(1) 行政思考決策輔助系統——成功人士必備工具
(2) 全能標貼製作器
(3) 大力水手——POPEYE
(4) 火樹銀花賀聖誕——平安夜
(5) 低解像圖輔助繪劃器
(6) 達爾文進化論的電腦模擬實踐
(7) 程式編輯器
(8) 磁碟檔案地圖
(9) 增加DOS的CATALOG暫停功能
(10) 新游標
(11) 40軌磁碟改動器
(12) 餅形圖
(13) 自己設信紙版頭
(14) 捉虫問題及答案
(15) 一行程式得獎作品

第12期程式磁碟內容：

① 家庭開支及個人消費管理系統
② 電子郵箱
③ 40軌程式COPY工具
④ 諾曼第登陸戰（模擬戰爭思考遊戲）
⑤ 工兵掘地雷
⑥ 新改良蛇之噩夢
⑦ 魔域迷宮（APPLESOFT版本）
⑧ 讓九個月大的嬰孩接觸電腦吧！
⑨ 有多項功能的AMPERSAND工具程式
⑩ 磁頭清潔器
⑪ 程式拯救者
⑫ TAB & XY
⑬ 符號逆輪型（RPN）計數機
⑭ 比RAM DISK更好的RAM咭控制中心
⑮ 圖形建造師
⑯ 捉虫訓練——SINE曲線
⑰ 一行程式

第14期程式磁碟14B內容：

DOS 3.3 FORMAT（SIDE A）
(A) 令蘋果發言的最佳工具MOCKING BOARD SPEECH WRITER
(B) 本港郵費計算諮詢系統
CP／M FORMAT（SIDE B）
(C) CP／M的密碼檔案
(D) 程式設計與FORTRAN所帶來的好處（雙精度計算例子）

精選程式磁碟

編號：CA00C
內容：私人會計系統
(1) 私人財政預算 （第七期）
(2) 私人收入紀錄 （第七期）
(3) 私人會計程式 （第六期）
CA00C 每張售價30元。

編號：CA00D
內容：工具程式集
(1) APPLESOFT程式偵錯系統（等六期）
(2) 將DOS移上RAM咭（第十期）
(3) MACRO 鍵盤 （第十期）
(4) RAM咭控制中心 （第十二期）
(5) 程式編輯器 （第十三期）

歡迎補購各期
程式磁碟磁帶或雜誌

調整售價啓事

電腦時代出版至今經已有十四期。在這年多以來，幸得讀者的支持，電腦時代才得已維持出版。我們衷心感謝。

長久以來，我們都抱有一個宗旨，就是要辦一本內容充實而有建設性的電腦雜誌，幫助讀者們加强對電腦的認識與及能發揮電腦的效能。我們盡量將雜誌及程式磁碟的訂價普及，務求每個擁有電腦的用家受惠。

毫無疑問，電腦時代能在芸芸衆多的電腦刋物之中豎立起本身的風格，並獲得大家的接受支持。雖然，電腦時代每期出版的頁數厚度不及其他電腦刋物，但論及內容的充實程度、題材的多元化發展，文章收藏的價値性，與及作者行文的可讀性兼且對讀者提供的服務，絕對可稱得上是最佳的一本。

電腦時代能成爲一本最受歡迎，全港銷量最高，最多人關心的電腦刋物，實非僥倖。當然，我們不會因此而自滿或驕傲；相反，我們不敢有半分的鬆懈，我們全體成員及幕後一羣作者會加倍努力，務求將內容辦得更加充實，以滿足讀者的需求，報答讀者對我們的支持。

在這年多以來，香港的物價不斷的上升，而我們盡量將雜誌的售價維持低水平，可是由於經營成本倍幅的增加，令我們感到非常吃力，我們不得不將雜誌售價調整以抵消經營成本的上漲壓力。電腦時代由第十五期開始，每本售價調整至10元，希望讀者能原諒我們的苦衷。並繼續加以支持。

至於程式磁碟則仍然保持舊價。爲了照顧忠實的長期讀者，我們決定將電腦時代現時的訂閱價格維持至本年二月底止，兼且依然提供特惠給雜誌與程式磁碟的訂戶，換言之，在本年二月底之前，訂閱電腦時代十二期只收九十六元（包括郵費），訂閱電腦時代十程式磁碟一年只收三百六十五元（比原價拆省20%）。

電腦時代出版社

第十五期內容預告

1. 電子訊息顯示板

各位看過一些LED 訊息顯示板，通常這些在商場或店舖門面出現的巨型訊息顯示板，可以將訊息不停的走動，重覆的顯示給顧客面前。我們令蘋果電腦都能擁有這項功能。讀者可以輸入最長達1000個字母（字母接近2吋高，在低解像頁顯示，遠距離也能看清楚），任意選定不同的顏色，在電腦螢幕上由右至左的出現及重覆最多達999次。而走動的速度是隨意調校。程式使用簡易，而用途則隨個人想像力而擴展。

2. 改動 JOYSTICK 成爲 AUTO FIRE

沒有AUTO FIRE掣的讀者，只要花數元便可將手上的JOYSTICK改爲AUTO FIRE ，步驟非常簡單。

3. 令蘋果運作 IBM 軟件的 AD 8088 咭

A D 8088咭是近期設計予APPLE 系列的高級週邊咭，主要的作用是提供予用家快速運算及發展專業軟件，發揮特殊功能和與IBM 軟件組相融之用。到底這塊高級週邊咭有可神奇之處，本文爲你深入的剖析。



電腦時代讀者服務部爲加强對讀者的服務質素，經已搬遷往交通更方便的旺角區，面積更寬闊的寫字樓。

新地址是旺角上海街438號同珍商業中心11字樓1104室（上海街／登打士街交界）。



歡迎讀者駕臨。

何必自己動手

免費 由專業人員上門爲你修理微型電腦



請即加入本公司之保養及免費維修服務

24小時熱綫電話服務：3-306839

微型電腦保養維修中心

ROM & RAM computer ltd

SHOP: 582–592, Nathan Road, Basement Shop No. B35 Sino Centre, Kowloon. Tel: 3-306839

OFFICE: Rm. 1905 Witty Coml. Bldg., 1A–1J, Tung Choi St., Kowloon. TEL: 3-848129

Telex: 57551 ROMHX

電腦時代讀者服務部與 COMPUHOUSE 聯合主辦

APPLE FAN CLUB

APPLE FAN CLUB（以下簡稱AFC）屬於一個不牟利性質的組織，旨在聯繫蘋果電腦用家，促進大家對電腦的應用。歡迎蘋果電腦用家參加。

AFC 提供的權利包括：

① 免費借用最新出版的蘋果電腦軟件。
② 以批發價錢供應會員各類磁碟、磁碟機、週邊咭。
③ 獲得每月三份的十日刊通訊錄，通訊錄報導各方面的最新軟件出版消息，完全免費。
④ 爲讀者搜羅各類的MANUAL 。
⑤ 供給各種新軟件的BACK UP或破解方法。
⑥ 會員可以安坐家中足不出戶取得所借用的軟件。

每個月，港九的高登及國都商場都推出有不少的新軟件，包括遊戲及工具軟件。我們將一一的搜羅這些軟件，存放在AFC內的SOFTWARE LIBRARY 內，以便供會員借用。每隔十天我們會寄出一份最新軟件出版的消息動態及名單，會員收到通訊錄後，可隨意選擇要借用的軟件，塡妥隨附的借用申請表格連同空磁碟（我們亦以批發價錢供應各類型磁碟）寄來AFC。我們收到借用表格後，在五個工作天內便會將會員要借用的軟件寄出。或者申請者可於指定時間到電腦時代讀者服務部攞取。每名會員每月可免費借用六個軟件，若要多借軟件，每個只收取象徵式一元的服務費（可用郵票代付），此外，别無其他收費。同時，我們亦提供有關的軟件的MANUAL ，我們只收回影印成本。

招收基本會員（名額有限　報名從速）

參加AFC資格：只要是蘋果電腦用家即可。

收費辦法：參加者每年須繳付年費180元。亦可分期繳交，辦法如下：

(1) 一次過繳交一年會費　$180　　(3) 分四期每季繳交一次　$60
(2) 分二期每半年繳交一次 $108　　(4) 分十二期每期繳交一次 $25

✱✱學生及長期訂戶可獲８折優待✱✱

各位只要塡妥下列表格，連同劃線支票寄來九龍中央郵政信箱CPO BOX 71193，電腦時代出版社收即可，支票抬頭塡寫〔COMPUTING AGE PUBLISHER〕，信封面請註明〔AFC〕。分期繳付會費的申請先繳付首期會費。

APPLE FAN CLUB 申請入會表格

申請者姓名：(英文) ______________________ (中文) ______________

通訊地址：(英文) ______________________________

____

(中文) ______________________________

____

通訊電話：______________ 職　業：______________

支票號碼及銀行：______________________ 申請日期 ______________

特惠優待證號碼✱ ________ 電腦時代訂閱戶✱ ________ COMPUSOFT磁碟月刊訂閱戶✱ ________

✱假如是訂閱戶請塡寫訂戶號碼

繳付會費辦法。

☐ 一次繳付　☐ 分二期繳付
☐ 分四期繳付　☐ 分十二期繳付

NO. ________
Official Use

✱✱香港以外地區申請者需加附郵費（空郵）40元，另外每次申請借用軟件需自付郵費。

學校電腦輔助教育

35 ■ 英文字義辨識練習

兒童電腦輔助教育

78 ■ 數恐龍學習加減運算

76 最新遊戲軟件評介

LUCIFERS REACM
KARATEKA
SNOOPY II
CONAN

BACK UP 資料館

45 KARATEKA 武士道的抄法
85 ■ HIDE：將程式隱藏

精品店

43 ■ 將程式轉為 TEXT 檔案來編輯
70 ■ 音量控制
86 ■ 一行程式參加得獎者
85 ■ 捉蟲訓練 DOS 指令的執行
45 RWTS 程式內部剖析

讀者園地

92 APPLE FAN CLUB 成立
3 COMPUSOFT磁碟月刊廣告介紹
11 業餘程式編寫員好消息
87 各期程式磁碟總目錄
84 更正

為了讓電腦用家將電腦應用於個人事業之上，我們由今期開始再增加一個新專欄〔中小型工商業應用〕刊登一些適合這類發展的文章或軟件。第一篇是介紹一個香港郵資計算諮詢系統，對於要經常寄信或貨辦的公司會有極大幫助。

應用軟件篇介紹的〔高解像圖魔術師〕能令你如玩魔術般弄出一些高解像圖花巧及特殊效果。其原理可應用於日後大型程式的編寫。

今期製作篇教大家改動CONTROLLER咭來BOOT起DRIVE 2。

COMPUHOUSE 與電腦時代出版社決定組織一個不牟利的電腦迷協會，為大家提供不少免費的福利及服務，詳情請閱本期廣告。

今期出版兩張程式磁碟，磁碟#14A大家仍可於各特約經銷處購買，程式磁碟#14B大家則要來本社服務部購買。售價一樣是29.90元。

■ 程式磁碟#14A
□ 程式磁碟#14B

特稿

□ 令蘋果發言的最佳工具 MOCKINGBOARD SPEECH WRITER 14
在機械語言中CALL BASIC 42
蘋果奧妙 44
改動VTOC獲得102個空SECTOR 82

高級篇

程式設計與FORTRAN所帶來的好處 32
□ MBASIC 檔案密碼鎖 36
□ MAILMERGE 的操作備忘（續完） 39

中小型工商業應用

□ 本港郵資計算諮詢系統 48

應用軟件庫

■ 高解像圖魔術師 23
■ 以軟件來達成的聲音合成器 50

小型製作專欄

BOOT 起 DRIVE 2 46

齊齊樂開心坊

■ 城市保衛戰 61
■ 最後一個立體打井遊戲 65
■ 月球一小步，人類一大步 69

蘋果工具箱

■ 磁碟度身定造器 71
■ SCREEN SAVER 40

軟件縱橫談

CP/M FORMAT 40 — CP/M 用戶可將磁碟容量提高15% 47
合初用者程度的工具 DOS BOSS 使用詳談 75

專欄

機械語言入門——機械語言對高解像度圖形的使用（上） 57
■ DOS 3.3 專欄——DOS 的快速存取 59

電腦時代

NO.14

本刊每月廿五號出版。一年十二期。全年訂閱費96港元（掛號每期加4元），雜誌連程式磁碟455港元（掛號每期加8元）。海外訂閱東南亞區空郵加10%郵費，其他地區加20%郵費。

◘出版：電腦時代出版社 ◘督印人：伍小華 ◘編輯：電腦時代編輯部 ◘軟件支援：COMPUHOUSE ◘通訊地址：九龍中央郵箱71193號 ◘通訊電話：3-312007 ◘承印者：永恒印刷公司 ◘發行人：同德書報社

電腦時代程式磁碟

每張只售HK$29.90

電腦時代每期都刊登多個精彩的程式，其中必然有你喜歡的。當你讀完文章興緻勃勃的坐下來欲進一步深入了解程式的精彩處時，看見那大堆冗長的列表，就甚麼興趣也沒有了！那麼，你爲何不購買一張「電腦時代程式磁碟」呢！

每期出版的電腦時代所登載的程式，均全部紀錄入一張高品質的磁碟上，並隨雜誌同時出版。爲省卻打入程式的麻煩，徒然浪費你寶貴的時間，免除了花精神捉錯打字的痛苦，今日就請就近的特約經銷處或電腦時代讀者服務部/COMPUHOUSE購備一張。

程式磁碟每張只售港幣 29.90 元。

第14 期程式磁碟A內容：

*All Programs are in Apple version

(1) 高解像圖魔術師
(2) 以軟件來達成的音响合成器
(3) SCREEN SAVER
(4) 磁碟度身定造器
(5) DOS 的快速存取
(6) 一行程式得獎作品選
(7) HIDE：將程式隱藏
(8) 英文字義辨讀練習
(9) 數恐龍學習加、減運算
(10) 宇宙城保衛戰（麥法基專欄）
(11) 最後一個立體打井遊戲
(12) 月球一小步，人類一大步
(13) 音量控制
(14) 將APPLESOFT程式轉爲TEXT檔案來編輯
(15) 捉虫訓練——DOS指令的執行

今期出版兩張程式磁碟，編號爲#14A 及#14B。14B爲CP／M格式，兩張皆是雙面灌錄。#14A可於各特約經銷處購買#14B則只在讀者服務部出售。詳情請閱書內磁碟目錄。

爲方便各區讀者購買程式磁碟，我們新增加多個特約經 銷處：以下爲各特約經銷處地址：

中環

百老滙攝影器材有限公司（第五店）
域多利皇后街 3 號（中環街市側）
電話：5－235513

亞當影音
士丹利街 28 A（陸羽茶樓側）
電話：5－240723

三聯書店讀者服務中心
香港中環域多利皇后街九號（中環街市側）
電話：5~250102

銅鑼灣

百老滙攝影器材有限公司（第十四店）
軒尼斯道 529號地下（三越百貨公司對面）
電話：5－770181

太古城

百老滙攝影器材有限公司（第七店）
太古城中心第 232店
電話：5－672006

西環

創業雜誌中心
西環屈地街22號地下　明愛中心對面
電話：5-478338

旺角

百老滙攝影器材有限公司　（第 1,3,9,10 店）
彌敦道 758號在下（凱聲戲院側）
電話：3－816931
彌敦道 746號地下（麗聲戲院側）
電話：3－943827
豉油街 26 號地下（近商務書局）
電話：3－884331
亞皆老街 60 B 地下（世界書局對面）
電話：3－956367

青年圖書有限公司
亞皆老街 80 號昌明大廈H座 1／F
（飛龍快餐店樓上）電話：3－918965

尖沙咀

智源書局
金巴利道 27—33號永利大厦 2字樓A座
電話：3 － 678482　3—678414

荔枝角

百老滙攝影器材有限公司（第二店）
美孚新邨第四期百老滙街 79 號（影都戲院側）
電話：3－7424923

深水埗

百老滙攝影器材有限公司（第十三店）
青山道148 號地下（華聲戲院對面）
電話：3－7295938

觀塘

百老滙攝影器材有限公司（第四、十二店）
物華街 3 號地下（皇上皇側）
電話：3－419513
同仁街 35 號地下（大家樂側）
電話：3－449940

荃灣

百老滙攝影器材有限公司（第十一店）
沙咀道 230 A（戴麟趾夫人健康院對面）
電話：0－438926

新增設新界屯門特約經銷商：

恒達時富僑公司
屯門華都花園地下商場 8 號室
電話：0-252577

教育專業人員協會（佐頓及北角分會）

電腦時代讀者服務部/ CompuHouse

地址：旺角上海街438號同珍商業中心11字樓1104室
（上海街／登打士街交界）　電話：3-312007

補購各期程式磁碟可前往讀者服務部／COMPU HOUSE

Apple is the trademark of Apple Computer Inc.